Matrix Concentration & Computational Linear Algebra

Short course at École Normale Supérieure, Paris, July 2019

Joel A. Tropp

Steele Family Professor of Applied & Computational Mathematics

California Institute of Technology

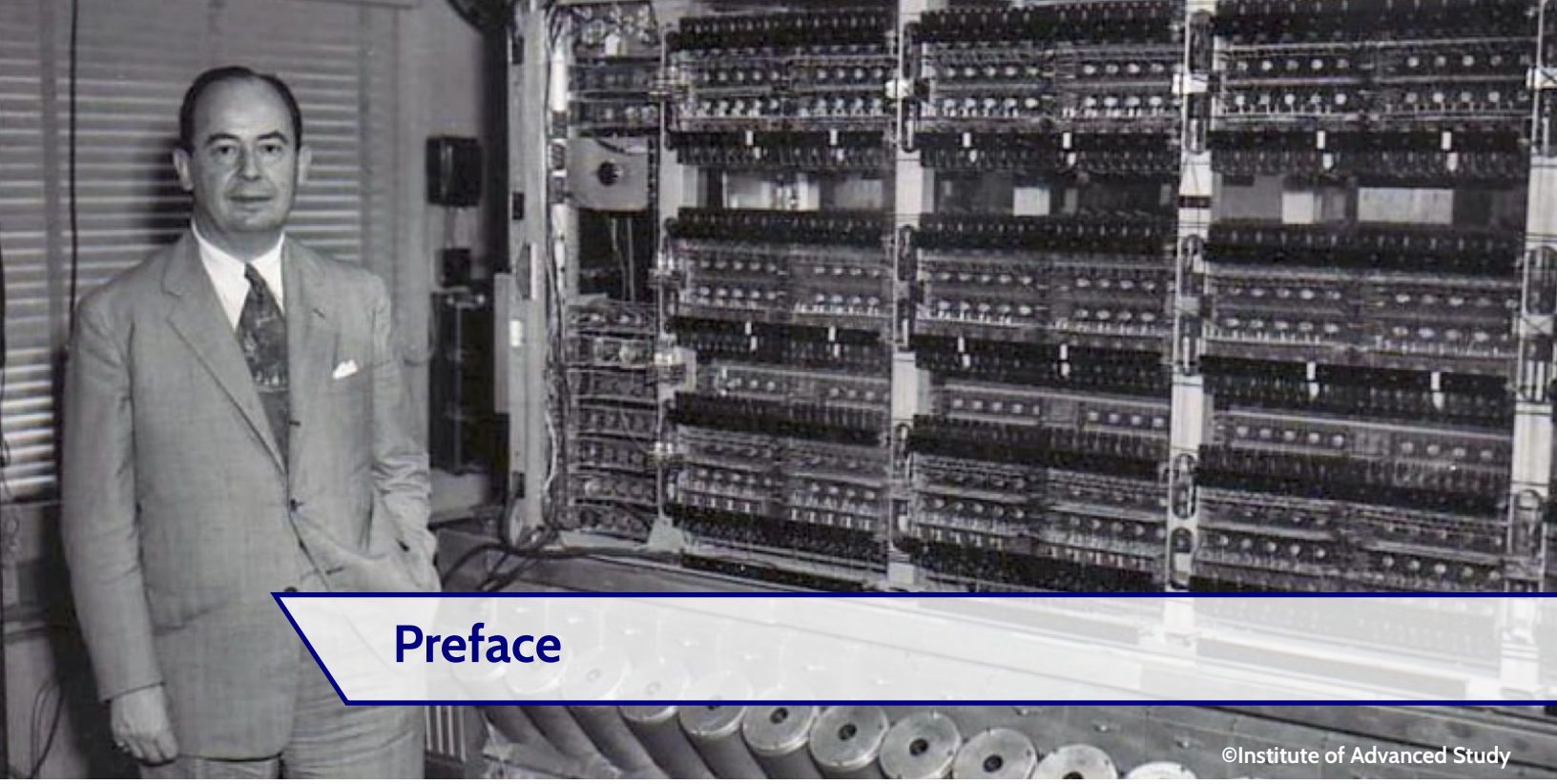
Typeset on February 23, 2021

Cite as:
Joel A. Tropp, *Matrix Concentration & Computational Linear Algebra*, Caltech CMS Lecture Notes 2019-01, Pasadena, July 2019.

Available from
http://resolver.caltech.edu/CaltechAUTHORS:20190715-125341188

Public domain

Contents



Preface

Over the last decade, random matrices have become ubiquitous in applied and computational mathematics. As this trend accelerates, more and more researchers must confront random matrices as part of their work. Classical random matrix theory can be difficult to use, and it is often silent about the questions that come up in modern applications. As a consequence, it has become imperative to develop new tools that are easy to use and that apply to a wide range of random matrices.

Matrix concentration inequalities

Matrix concentration inequalities are among the most popular of these new methods. For a self-adjoint random matrix $\boldsymbol{Y}$ with expectation $\mathbb{E}\,\boldsymbol{Y}$, matrix concentration theorems provide probabilistic bounds on quantities like

$$\|\boldsymbol{Y} - \mathbb{E}\,\boldsymbol{Y}\|.$$

The symbol $\|\cdot\|$ always refers the spectral norm, also known as the ℓ_2 operator norm. Bounds of this form give us a lot of information about how the random matrix $\boldsymbol{Y}$ is related to its expectation $\mathbb{E}\,\boldsymbol{Y}$. In particular,

- Each linear functional of $\boldsymbol{Y}$ is close to the same linear functional of $\mathbb{E}\,\boldsymbol{Y}$.
- Each eigenvalue of $\boldsymbol{Y}$ is close to the corresponding eigenvalue of $\mathbb{E}\,\boldsymbol{Y}$.
- Each eigenvector of $\boldsymbol{Y}$ is close to the corresponding eigenvector of $\mathbb{E}\,\boldsymbol{Y}$ when the eigenvalue is isolated from the rest of the spectrum.
- We can bound the expected norm of the random matrix:

$$\|\boldsymbol{Y}\| = \|\mathbb{E}\,\boldsymbol{Y}\| \pm \|\boldsymbol{Y} - \mathbb{E}\,\boldsymbol{Y}\|.$$

The last point is, perhaps, the most interesting. Indeed, norm bounds for random matrices are quite valuable by themselves, and they used to be rather hard to obtain before the matrix concentration technology was developed.

Matrix concentration results for self-adjoint random matrices also have formal consequences for rectangular random matrices. We will focus on the self-adjoint case because it is more fundamental, and it already supports many fascinating applications.

Random matrix models

Without additional information about the random matrix $\boldsymbol{Y}$, we cannot hope to say anything interesting. This work treats two basic, but very fruitful, models for the random matrix.

First, the *independent sum model* posits that

$$\boldsymbol{Y} = \sum\nolimits_{i=1}^{n} \boldsymbol{X}_i \quad \text{where } \{\boldsymbol{X}_i\} \text{ is statistically independent.}$$

This model captures a wide range of examples. The most classical is the sample covariance matrix; see [Tro15, Chap. 1] for discussion and analysis. In this course, we will explore more modern examples from machine learning, quantum information theory, and combinatorics.

Second, we will consider the *matrix martingale model*, where

$$\boldsymbol{Y}_k = \sum\nolimits_{i=1}^{k} \boldsymbol{X}_i \quad \text{is a martingale.}$$

This model offers a powerful lens for studying the behavior of iterative randomized algorithms in linear algebra. The main purpose of this course is to show how concentration for matrix martingales supported the development and analysis of an efficient algorithm for solving graph Laplacian linear systems.

Other applications of matrix concentration

Matrix concentration tools have already found a place in many areas of the mathematical sciences, including

- numerical linear algebra [Tro11b]
- numerical analysis [MB17]
- uncertainty quantification [CG14]
- statistics [Kol11]
- econometrics [CC13]
- approximation theory [CDL13]
- sampling theory [BG13]
- machine learning [DKC13; Lop+14]
- learning theory [FSV12; MKR12]
- math signal processing [Che+14]
- optimization [CSW12]
- graphics and vision [HCG14]
- quantum information [Hol12]
- algorithms [HO14; Kyn17]
- combinatorics [Oli10]
- *et cetera*.

These references are chosen more or less at random from a long menu of possibilities. See the monograph [Tro15] for an overview of the main results on matrix concentration, many detailed applications, and additional background references. Other recommendations for further reading appear at the end of these notes.

About this course

These lecture notes were written to support the short course

Matrix Concentration & Computational Linear Algebra

delivered by the author at École Normale Supérieure in Paris from 1–5 July 2019 as part of the summer school "High-dimensional probability and algorithms."

The aim of this course is to present some practical computational applications of matrix concentration. Lecture 1 provides a brief treatment of the matrix Bernstein inequality, which is the most valuable single result about matrix concentration. We apply this result to study several empirical matrix approximations:

- Random feature approximation of a kernel matrix (Lecture 2).
- Linear estimators for quantum state tomography (Lecture 3).
- Sparse approximation of a combinatorial graph (Lecture 5).

Our primary goal is to develop a complete treatment of a near-linear time algorithm for solving a linear system in a graph Laplacian matrix. This remarkable algorithm was developed by Rasmus Kyng and Sushant Sachdeva [KS16], following earlier work [Kyn+16] by Dan Spielman's group. The algorithm closely resembles the classic incomplete Cholesky decomposition, and I believe that it is likely to have an impact on computational practice. Our presentation of this result takes place in steps:

- Harmonic analysis on graphs (Lecture 4).
- Interpretation of graphs as resistor networks (Lecture 5).
- Cholesky factorization of a graph Laplacian (Lecture 6).
- Theory of matrix martingales (Lecture 7).
- The SparseCholesky algorithm (Lecture 8).

In my opinion, the SparseCholesky algorithm is the most spectacular application of matrix concentration. I doubt that it could have been developed before the foundations of matrix concentration were in place.

Prerequisites

Since the audience of this short course consists primarily of French graduate students and researchers, I have assumed a moderate level of mathematical and computational preparation:

- Intermediate linear algebra [Axl15], including experience with positive-semidefinite matrices and the semidefinite order [Bha97; Bha07].
- Elementary numerical linear algebra [TB97], including Cholesky decomposition, solution of triangular systems, and the conjugate gradient algorithm.
- Intermediate probability, including basic scalar concentration inequalities [BLM13] and real-valued discrete-time martingales [Wil91].
- Elementary spectral graph theory [Spi12].

We will develop the background for most of the applications in sufficient detail that no additional preparation is needed.

Caveat lector

These notes do not meet the standard of a scholarly publication. Here are some issues that you should be aware of.

- Some of the text has been copied and pasted directly from my own published work (Lectures 1, 2, and 7).
- The notes for Lecture 3 were written primarily by Richard Kueng on the basis of our joint work [Guh+18]. I take responsibility for any mistakes that appear.
- The treatment of graph theory is my interpretation of Dan Spielman's lecture notes [Spi] and Rasmus Kyng's dissertation [Kyn17]. Any errors are mine.
- Owing to the varied provenance of the material, the mathematical notation may not be fully consistent among different lectures.
- I have tried to provide citations for the main results, but these notes are largely devoid of references, historical background, and context.
- These notes have only received a cursory proofreading.

Why is there a photo of von Neumann?

You may be wondering why John von Neumann greets you at the door of this Preface. In 1947, von Neumann and Goldstine [NG47] developed the foundations for rounding error analysis. They formulated Gaussian elimination and Cholesky decomposition as triangular matrix factorizations. They showed how to analyze the numerical properties of the linear system solver based on this approach. In a 1951 follow-up paper [GN51], they proposed a random matrix model for the rounding errors in these computations. See [Grc11] for a gloss on this research.

The landmark papers of von Neumann and Goldstine are among the earliest works on solving linear systems on a computer, and they are the first to bring random matrix theory in contact with computational linear algebra. I cannot think of a more suitable genie to inhabit these notes.

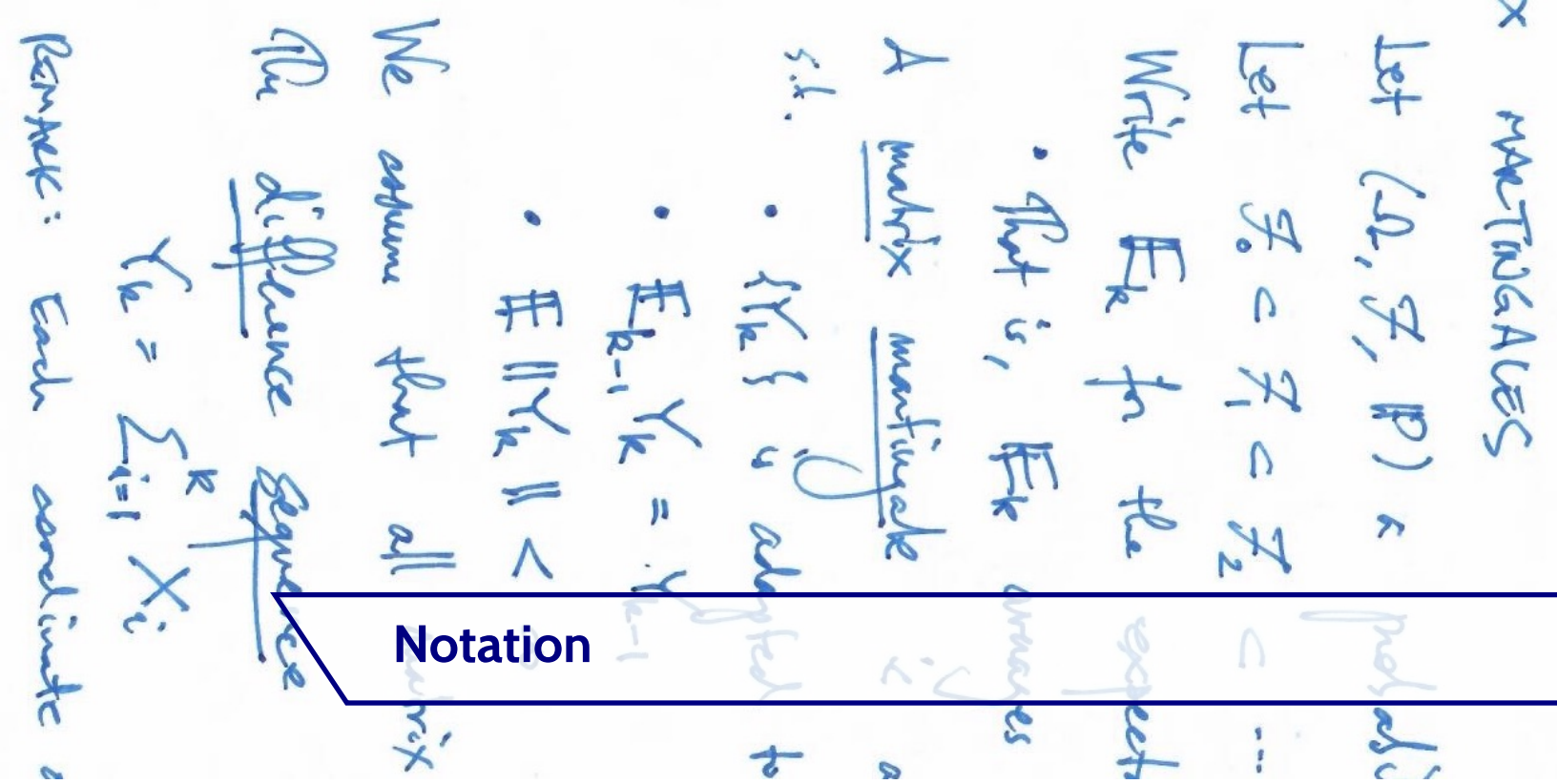
Acknowledgments

The summer school "High-dimensional probability and algorithms" was funded by Université PSL and CNRS. I would like to thank the organizers, Claire Boyer, Djalil Chafaï, and Joseph Lehec, for an engaging week. Additional funding for my research and this course was provided by ONR Awards N00014-17-12146 and N00014-18-12363.

The computational cost of the SparseCholesky algorithm was miscalculated in the original manuscript because of stray parentheses; the costs are slightly higher than reported. This version corrects the error, which was reported by Rasmus Kyng.

Joel A. Tropp
jtropp@cms.caltech.edu
http://users.cms.caltech.edu/~jtropp

Steele Family Professor of Applied & Computational Mathematics
California Institute of Technology
Pasadena, California
July 2019



Notation

I have selected notation that is common in the linear algebra and probability literature. I have tried to been consistent in using the symbols that are presented below. There are some minor variations in different lectures, including the letter that indicates the dimension of a matrix and the indexing of sums.

Linear algebra

We work in a real or complex linear space. The letters d and n (and occasionally others) are used to denote the dimension of this space, which is always finite. For example, we write $\mathbb{R}^d$ or $\mathbb{C}^n$. Matrix concentration results apply equally in the real and complex setting. We may write $\mathbb{F}$ to refer to either field, or we may omit the field entirely.

We use the delta notation for standard basis vectors: $\boldsymbol{\delta}_i$ has a one in the ith coordinate and zeros elsewhere. The vector $\mathbf{1}$ has ones in each entry. The dimension of these vectors is determined by context.

The symbol * denotes the (conjugate) transpose of a vector or a matrix. We equip $\mathbb{F}^d$ with the standard inner product $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \boldsymbol{x}^* \boldsymbol{y}$. The inner product generates the Euclidean norm $\|\boldsymbol{x}\|^2 = \langle \boldsymbol{x}, \boldsymbol{x} \rangle$.

We write $\mathbb{H}_d(\mathbb{F})$ for the real-linear space of $d \times d$ self-adjoint matrices with entries in the field $\mathbb{F}$. Recall that a matrix is self-adjoint when $\boldsymbol{A} = \boldsymbol{A}^*$. We equip the space $\mathbb{H}_d$ with the trace inner product $\langle \boldsymbol{X}, \boldsymbol{Y} \rangle = \operatorname{tr}(\boldsymbol{XY})$, which generates the Frobenius norm $\|\boldsymbol{X}\|_{\mathrm{F}}^2 = \langle \boldsymbol{X}, \boldsymbol{X} \rangle$. The map $\operatorname{tr}[\cdot]$ returns the trace of a square matrix; we instate the convention that nonlinear functions bind before the trace.

A self-adjoint matrix with dimension d has d real eigenvalues, with an associated orthonormal set of d eigenvectors. The maps $\lambda_{\min}(\cdot)$ and $\lambda_{\max}(\cdot)$ return the minimum and maximum eigenvalues of a self-adjoint matrix. The symbol $\mathbf{I}$ denotes the identity matrix; its dimensions are determined by context.

A self-adjoint matrix is *positive semidefinite* (psd) if its eigenvalues are nonnegative; a self-adjoint matrix is *positive definite* (pd) if its eigenvalues are positive. The symbol $\preccurlyeq$ refers to the psd order: $\boldsymbol{A} \preccurlyeq \boldsymbol{H}$ if and only if $\boldsymbol{H} - \boldsymbol{A}$ is psd.

We define a *standard matrix function* on a self-adjoint matrix using the eigenvalue decomposition. For any $f : \mathbb{R} \to \mathbb{R}$,

$$\boldsymbol{A} = \sum\nolimits_{i=1}^{n} \lambda_i \, \boldsymbol{u}_i \boldsymbol{u}_i^* \quad \text{implies} \quad f(\boldsymbol{A}) = \sum\nolimits_{i=1}^{n} f(\lambda_i) \, \boldsymbol{u}_i \boldsymbol{u}_i^*.$$

When we apply a real function to a self-adjoint matrix, we are always referring to the associated standard matrix function. In particular, we often encounter powers, exponentials, and logarithms.

Occasionally, we need the linear space $\mathbb{M}^{d_1 \times d_2}(\mathbb{F})$ of $d_1 \times d_2$ matrices over the field $\mathbb{F}$. In this context, the symbol $\|\cdot\|$ also refers to the ℓ_2 operator norm.

We write lin for the linear hull of a family of vectors. The operators range and null extract the range and null space of a matrix. The operator † extracts the pseudoinverse.

Probability

The map $\mathbb{P}\{\cdot\}$ returns the probability of an event. The operator $\mathbb{E}[\cdot]$ returns the expectation of a random variable taking values in a linear space. We only include the brackets when it is necessary for clarity, and we impose the convention that nonlinear functions bind before the expectation.

Graphs

A multigraph G has a ground set V of n vertices. A multiedge is an undirected pair $e = uv = \{u, v\}$ of vertices. A multigraph involves a set E of m multiedges, which may involve many edges connecting the same pair of vertices. The absolute value $|\cdot|$ returns the cardinality of a set of vertices or a set of edges.

We write $\mathbb{R}^{\mathsf{V}}$ for the set of real-valued functions on the set V of vertices. The symbol $\mathbb{H}_{\mathsf{V}}$ refers to the linear space of (real) self-adjoint matrices acting on $\mathbb{R}^{\mathsf{V}}$. We may identify these linear spaces with $\mathbb{R}^n$ and $\mathbb{H}_n(\mathbb{R})$.

The notation $u \sim v$ means that two vertices are neighbors. The notations $u \in e$ and $e \ni u$ both mean that the multiedge e is incident on (i.e., contains) the vertex u.

The degree $\deg(u, \mathsf{G})$ of a vertex u in a multigraph G is the total number of multiedges incident on u. The total weight $w_{\mathsf{G}}(u)$ of a vertex u is the sum of the weights of the multiedges incident on u.

We reserve the letter $\boldsymbol{L}$ for the Laplacian matrix of the multigraph G. The symbol $\boldsymbol{\Phi}$ denotes the normalizing map associated with this Laplacian:

$$\boldsymbol{\Phi}(\boldsymbol{M}) = (\boldsymbol{L}^\dagger)^{1/2} \boldsymbol{M} (\boldsymbol{L}^\dagger)^{1/2}.$$

The exponent $^{1/2}$ extracts the unique psd square root of a psd matrix. The number $\varrho(u, v)$ is the effective resistance between vertices u and v.

Order notation

We use the familiar order notation from computer science. The symbol $\Theta(\cdot)$ refers to asymptotic equality. The symbol $O(\cdot)$ refers to an asymptotic upper bound.

1. Matrix Concentration

Most of the text in this lecture is copied from my monograph [Tro15, Chaps. 3, 5, 6].

This lecture contains the analysis that delivers exponential matrix concentration inequalities. The approach that we take can be viewed as a matrix extension of the Laplace transform method, sometimes referred to as the "Bernstein trick." In the scalar setting, this trick (*soi disant*) is one of the most basic and successful paths to reach concentration inequalities for sums of independent random variables. It turns out that there is a very satisfactory version of this argument that applies to sums of independent random matrices. In the general setting, however, we must invest more care and wield sharper tools to execute this technique.

1.1 The matrix Laplace transform method

In the scalar setting, the Laplace transform method allows us to obtain tail bounds for a random variable in terms of its mgf. The starting point for our theory is the observation that a similar result holds in the matrix setting.

1.1.1 Tail bounds

First, we introduce the Laplace transform method for bounding the extreme eigenvalues of a self-adjoint matrix. This approach to matrix concentration was proposed by Ahlswede & Winter [AW02]. We present a formulation and proof developed by Roberto Oliveira [Oli10].

Proposition 1.1 (Tail bounds for eigenvalues). Let $\boldsymbol{Y}$ be a random self-adjoint matrix. For

all $t \in \mathbb{R}$,

$$\mathbb{P}\{\lambda_{\max}(\boldsymbol{Y}) \geq t\} \leq \inf_{\theta > 0} \mathrm{e}^{-\theta t}\ \mathbb{E}\,\mathrm{tr}\,\mathrm{e}^{\theta \boldsymbol{Y}}; \tag{1.1}$$

$$\mathbb{P}\{\lambda_{\min}(\boldsymbol{Y}) \leq t\} \leq \inf_{\theta < 0} \mathrm{e}^{-\theta t}\ \mathbb{E}\,\mathrm{tr}\,\mathrm{e}^{\theta \boldsymbol{Y}}. \tag{1.2}$$

In words, we can control the tail probabilities of the extreme eigenvalues of a random matrix by producing a bound for the *trace* of the matrix mgf. The proof of this fact parallels the classical argument, but there is a twist.

Proof. We begin with (1.1). Fix a positive number θ, and observe that

$$\mathbb{P}\{\lambda_{\max}(\boldsymbol{Y}) \geq t\} = \mathbb{P}\left\{\mathrm{e}^{\theta \lambda_{\max}(\boldsymbol{Y})} \geq \mathrm{e}^{\theta t}\right\} \leq \mathrm{e}^{-\theta t}\ \mathbb{E}\,\mathrm{e}^{\theta \lambda_{\max}(\boldsymbol{Y})}.$$

The first identity holds because $a \mapsto \mathrm{e}^{\theta a}$ is a monotone increasing function, so the event does not change under the mapping. The second relation is Markov's inequality. To control the exponential, note that

$$\mathrm{e}^{\theta \lambda_{\max}(\boldsymbol{Y})} = \mathrm{e}^{\lambda_{\max}(\theta \boldsymbol{Y})} = \lambda_{\max}\big(\mathrm{e}^{\theta \boldsymbol{Y}}\big) \leq \mathrm{tr}\,\mathrm{e}^{\theta \boldsymbol{Y}}. \tag{1.3}$$

The first identity holds because the maximum eigenvalue is a positive-homogeneous map. The second depends on the spectral mapping theorem. The inequality follows because the exponential of an self-adjoint matrix is positive definite. The maximum eigenvalue of a positive-definite matrix is dominated by the trace. Combine the latter two displays to reach

$$\mathbb{P}\{\lambda_{\max}(\boldsymbol{Y}) \geq t\} \leq \mathrm{e}^{-\theta t}\ \mathbb{E}\,\mathrm{tr}\,\mathrm{e}^{\theta \boldsymbol{Y}}.$$

This inequality is valid for any positive θ, so we may take an infimum to achieve the tightest possible bound.

To prove (1.2), we use a similar approach. Fix a negative number θ, and calculate that

$$\mathbb{P}\{\lambda_{\min}(\boldsymbol{Y}) \leq t\} = \mathbb{P}\left\{\mathrm{e}^{\theta \lambda_{\min}(\boldsymbol{Y})} \geq \mathrm{e}^{\theta t}\right\} \leq \mathrm{e}^{-\theta t}\ \mathbb{E}\,\mathrm{e}^{\theta \lambda_{\min}(\boldsymbol{Y})} = \mathrm{e}^{-\theta t}\ \mathbb{E}\,\mathrm{e}^{\lambda_{\max}(\theta \boldsymbol{Y})}.$$

The function $a \mapsto \mathrm{e}^{\theta a}$ reverses the inequality in the event because it is monotone decreasing. The last identity depends on the relationship between minimum and maximum eigenvalues. Finally, we introduce the inequality (1.3) for the trace exponential and minimize over negative values of θ. ∎

In the proof of Proposition 1.1, it may seem crude to bound the maximum eigenvalue by the trace. It turns out that, at most, this estimate results in a loss of a factor that is logarithmic in the dimension of the matrix. At the same time, our maneuver allows us to exploit some amazing convexity properties of the trace exponential.

1.1.2 Expectation bounds

We can adapt the proof of Proposition 1.1 to obtain bounds for the expectation of the maximum eigenvalue of a random self-adjoint matrix. This argument is somewhat less interesting in the scalar setting, where it states that the exponential mean of a random variable is an upper bound for the arithmetic mean.

Proposition 1.2 (Expectation bounds for eigenvalues). Let $\boldsymbol{Y}$ be a random self-adjoint matrix. Then

$$\mathbb{E}\,\lambda_{\max}(\boldsymbol{Y}) \le \inf_{\theta>0} \frac{1}{\theta} \log \mathbb{E}\,\operatorname{tr} \mathrm{e}^{\theta \boldsymbol{Y}}; \tag{1.4}$$

$$\mathbb{E}\,\lambda_{\min}(\boldsymbol{Y}) \ge \sup_{\theta<0} \frac{1}{\theta} \log \mathbb{E}\,\operatorname{tr} \mathrm{e}^{\theta \boldsymbol{Y}}. \tag{1.5}$$

Proof. We establish the bound (1.4); the proof of (1.5) is quite similar. Fix a positive number θ, and calculate that

$$\begin{aligned}\mathbb{E}\,\lambda_{\max}(\boldsymbol{Y}) &= \frac{1}{\theta}\,\mathbb{E}\,\log \mathrm{e}^{\lambda_{\max}(\theta \boldsymbol{Y})} \le \frac{1}{\theta} \log \mathbb{E}\,\mathrm{e}^{\lambda_{\max}(\theta \boldsymbol{Y})} \\ &= \frac{1}{\theta} \log \mathbb{E}\,\lambda_{\max}\bigl(\mathrm{e}^{\theta \boldsymbol{Y}}\bigr) \le \frac{1}{\theta} \log \mathbb{E}\,\operatorname{tr} \mathrm{e}^{\theta \boldsymbol{Y}}.\end{aligned}$$

The first identity holds because the maximum eigenvalue is a positive-homogeneous map. The second relation is Jensen's inequality. The third follows when we use the spectral mapping theorem to draw the exponential inside the eigenvalue map. The final inequality depends on the fact that the trace of a positive-definite matrix dominates the maximum eigenvalue. ∎

1.2 Matrix moments and cumulants

At the heart of the Laplace transform method are the moment generating function (mgf) and the cumulant generating function (cgf) of a random variable. In this section, we define these functions rigorously, and we explore some of their properties.

1.2.1 The matrix mgf and cgf

We begin by presenting matrix versions of the mgf and cgf.

Definition 1.3 (Matrix mgf and cgf). Let $\boldsymbol{X}$ be a random self-adjoint matrix. The *matrix moment generating function* $\boldsymbol{M}_{\boldsymbol{X}}$ and the *matrix cumulant generating function* $\boldsymbol{\Xi}_{\boldsymbol{X}}$ are given by

$$\boldsymbol{M}_{\boldsymbol{X}}(\theta) = \mathbb{E}\,\mathrm{e}^{\theta \boldsymbol{X}} \quad \text{and} \quad \boldsymbol{\Xi}_{\boldsymbol{X}}(\theta) = \log \mathbb{E}\,\mathrm{e}^{\theta \boldsymbol{X}} \quad \text{for } \theta \in \mathbb{R}. \tag{1.6}$$

Note that the expectations may not exist for all values of θ.

The matrix mgf $\boldsymbol{M}_{\boldsymbol{X}}$ and matrix cgf $\boldsymbol{\Xi}_{\boldsymbol{X}}$ contain information about the distribution of the random matrix $\boldsymbol{X}$, including its mean and variance. Propositions 1.1 and 1.2 show how to exploit the data encoded in these functions to control the eigenvalues.

Let us dilate on Definition 1.3. Observe that the matrix mgf and cgf have formal power series expansions:

$$\boldsymbol{M}_{\boldsymbol{X}}(\theta) = \mathbf{I} + \sum_{q=1}^{\infty} \frac{\theta^q}{q!} (\mathbb{E}\, \boldsymbol{X}^q) \quad \text{and} \quad \boldsymbol{\Xi}_{\boldsymbol{X}}(\theta) = \sum_{q=1}^{\infty} \frac{\theta^q}{q!} \boldsymbol{\Psi}_q.$$

We call the coefficients $\mathbb{E}\, \boldsymbol{X}^q$ *matrix moments*, and we refer to $\boldsymbol{\Psi}_q$ as a *matrix cumulant*. The matrix cumulant $\boldsymbol{\Psi}_q$ has a formal expression as a (noncommutative) polynomial in the matrix moments up to order q. In particular, the first cumulant is the mean and the second cumulant is the variance:

$$\boldsymbol{\Psi}_1 = \mathbb{E}\, \boldsymbol{X} \quad \text{and} \quad \boldsymbol{\Psi}_2 = \mathbb{E}\, \boldsymbol{X}^2 - (\mathbb{E}\, \boldsymbol{X})^2.$$

Higher-order matrix cumulants are harder to write down and interpret.

1.2.2 The failure of the matrix mgf

We would like the use the Laplace transform bounds from Section 1.1 to study a sum of independent random matrices. In the scalar setting, the Laplace transform method is effective for studying independent sums because the mgf and the cgf decompose. In the matrix case, the situation is more subtle, and the goal of this section is to indicate where things go awry.

Consider an independent sequence $\{X_k\}$ of real random variables. The mgf of the sum satisfies a multiplication rule:

$$M_{(\sum_k X_k)}(\theta) = \mathbb{E} \exp\left(\sum\nolimits_k \theta X_k\right) = \mathbb{E} \prod\nolimits_k \mathrm{e}^{\theta X_k} = \prod\nolimits_k \mathbb{E}\, \mathrm{e}^{\theta X_k} = \prod\nolimits_k M_{X_k}(\theta). \tag{1.7}$$

The first identity is the definition of an mgf. The second relation holds because the exponential map converts a sum of real scalars to a product, and the third relation requires the independence of random variables. The last identity, again, is the definition.

At first, we might imagine that a similar relationship holds for the matrix mgf. Consider an independent sequence $\{\boldsymbol{X}_k\}$ of random self-adjoint matrices. Perhaps,

$$\boldsymbol{M}_{(\sum_k \boldsymbol{X}_k)}(\theta) \quad \stackrel{?}{=} \quad \prod\nolimits_k \boldsymbol{M}_{\boldsymbol{X}_k}(\theta). \tag{1.8}$$

Unfortunately, this hope shatters when we subject it to interrogation.

It is not hard to find the reason that (1.8) fails. The identity (1.7) depends on the fact that the scalar exponential converts a sum into a product. In contrast, for self-adjoint matrices,

$$\mathrm{e}^{\boldsymbol{A}+\boldsymbol{H}} \neq \mathrm{e}^{\boldsymbol{A}} \mathrm{e}^{\boldsymbol{H}} \quad \text{unless } \boldsymbol{A} \text{ and } \boldsymbol{H} \text{ commute.}$$

If we introduce the trace, the situation improves somewhat:

$$\operatorname{tr} \mathrm{e}^{\boldsymbol{A}+\boldsymbol{H}} \leq \operatorname{tr} \mathrm{e}^{\boldsymbol{A}} \mathrm{e}^{\boldsymbol{H}} \quad \text{for all self-adjoint } \boldsymbol{A}, \boldsymbol{H}. \tag{1.9}$$

The result (1.9) is known as the Golden–Thompson inequality, a famous theorem from statistical physics. Unfortunately, the analogous bound may fail for three matrices:

$$\operatorname{tr} \mathrm{e}^{\boldsymbol{A}+\boldsymbol{H}+\boldsymbol{T}} \not\leq \operatorname{tr} \mathrm{e}^{\boldsymbol{A}} \mathrm{e}^{\boldsymbol{H}} \mathrm{e}^{\boldsymbol{T}} \quad \text{for certain self-adjoint } \boldsymbol{A}, \boldsymbol{H}, \boldsymbol{T}.$$

It seems that we have reached an impasse.

What if we consider the cgf instead? The cgf of a sum of independent real random variables satisfies an addition rule:

$$\Xi_{(\sum_k X_k)}(\theta) = \log \mathbb{E} \exp\left(\sum\nolimits_k \theta X_k\right) = \log \prod\nolimits_k \mathbb{E}\, \mathrm{e}^{\theta X_k} = \sum\nolimits_k \Xi_{X_k}(\theta). \tag{1.10}$$

The relation (1.10) follows when we extract the logarithm of the multiplication rule (1.7). This result looks like a more promising candidate for generalization because a sum of self-adjoint matrices remains self-adjoint. We might hope that

$$\boldsymbol{\Xi}_{(\sum_k \boldsymbol{X}_k)}(\theta) \quad \stackrel{?}{=} \quad \sum\nolimits_k \boldsymbol{\Xi}_{\boldsymbol{X}_k}(\theta).$$

As stated, this putative identity also fails. Nevertheless, the addition rule (1.10) admits a very satisfactory extension to matrices. In contrast to the scalar case, the proof involves much deeper considerations.

1.2.3 A theorem of Lieb

To find the appropriate generalization of the addition rule for cgfs, we turn to the literature on matrix analysis. Here, we discover a famous result of Elliott Lieb on the convexity properties of the trace exponential function.

Theorem 1.4 (Lieb, 1973). Fix a self-adjoint matrix $\boldsymbol{H}$ with dimension d. The function

$$\boldsymbol{A} \longmapsto \operatorname{tr} \exp(\boldsymbol{H} + \log \boldsymbol{A})$$

is a concave map on the convex cone of $d \times d$ positive-definite matrices.

In the scalar case, the analogous function $a \mapsto \exp(h + \log a)$ is linear, so this result describes a new type of phenomenon that emerges when we move to the matrix setting. See [Tro15, Chap. 8] for a complete proof of Theorem 1.4 from first principles.

Lieb's theorem is valuable to us because the Laplace transform bounds from Section 1.1 involve the trace exponential function. To highlight the connection, let us rephrase Theorem 1.4 in probabilistic terms.

Corollary 1.5 (Tropp, 2010). Let $\boldsymbol{H}$ be a fixed self-adjoint matrix, and let $\boldsymbol{X}$ be a random self-adjoint matrix of the same dimension. Then

$$\mathbb{E} \operatorname{tr} \exp(\boldsymbol{H} + \boldsymbol{X}) \le \operatorname{tr} \exp\left(\boldsymbol{H} + \log \mathbb{E}\, \mathrm{e}^{\boldsymbol{X}}\right).$$

Proof. Introduce the random matrix $\boldsymbol{Y} = \mathrm{e}^{\boldsymbol{X}}$. Then

$$\begin{aligned} \mathbb{E} \operatorname{tr} \exp(\boldsymbol{H} + \boldsymbol{X}) &= \mathbb{E} \operatorname{tr} \exp(\boldsymbol{H} + \log(\boldsymbol{Y})) \\ &\le \operatorname{tr} \exp(\boldsymbol{H} + \log(\mathbb{E}\, \boldsymbol{Y})) = \operatorname{tr} \exp\left(\boldsymbol{H} + \log \mathbb{E}\, \mathrm{e}^{\boldsymbol{X}}\right). \end{aligned}$$

The first identity follows from the interpretation of the matrix logarithm as the functional inverse of the matrix exponential for positive-definite matrices. Theorem 1.4 shows that the trace function is concave in $\boldsymbol{Y}$, so Jensen's inequality allows us to draw the expectation inside the function. ∎

1.2.4 Subadditivity of the matrix cgf

We are now prepared to generalize the addition rule (1.10) for scalar cgfs to the matrix setting. The following result is fundamental to our approach to random matrices.

Lemma 1.6 (Subadditivity of matrix cgfs). Consider a finite sequence $\{\boldsymbol{X}_k\}$ of independent, random, self-adjoint matrices of the same dimension. Then

$$\mathbb{E}\,\operatorname{tr}\exp\left(\sum\nolimits_k \theta\boldsymbol{X}_k\right) \le \operatorname{tr}\exp\left(\sum\nolimits_k \log\mathbb{E}\,\mathrm{e}^{\theta\boldsymbol{X}_k}\right) \quad \text{for } \theta\in\mathbb{R}. \tag{1.11}$$

Equivalently,

$$\operatorname{tr}\exp\left(\boldsymbol{\Xi}_{(\sum_k \boldsymbol{X}_k)}(\theta)\right) \le \operatorname{tr}\exp\left(\sum\nolimits_k \boldsymbol{\Xi}_{\boldsymbol{X}_k}(\theta)\right) \quad \text{for } \theta\in\mathbb{R}. \tag{1.12}$$

The parallel between the additivity rule (1.10) and the subadditivity rule (1.12) is striking. With our level of preparation, it is easy to prove this result. We just apply the bound from Corollary 1.5 repeatedly.

Proof. Without loss of generality, we assume that $\theta = 1$ by absorbing the parameter into the random matrices. Let $\mathbb{E}_k$ denote the expectation with respect to $\boldsymbol{X}_k$, the remaining random matrices held fixed. Abbreviate

$$\boldsymbol{\Xi}_k = \log\mathbb{E}_k\,\mathrm{e}^{\boldsymbol{X}_k} = \log\mathbb{E}\,\mathrm{e}^{\boldsymbol{X}_k}.$$

We may calculate that

$$\begin{aligned}
\mathbb{E}\,\operatorname{tr}\exp\left(\sum\nolimits_{k=1}^{n}\boldsymbol{X}_k\right) &= \mathbb{E}\,\mathbb{E}_n\,\operatorname{tr}\exp\left(\sum\nolimits_{k=1}^{n-1}\boldsymbol{X}_k + \boldsymbol{X}_n\right)\\
&\le \mathbb{E}\,\operatorname{tr}\exp\left(\sum\nolimits_{k=1}^{n-1}\boldsymbol{X}_k + \log\left(\mathbb{E}_n\,\mathrm{e}^{\boldsymbol{X}_n}\right)\right)\\
&= \mathbb{E}\,\mathbb{E}_{n-1}\,\operatorname{tr}\exp\left(\sum\nolimits_{k=1}^{n-2}\boldsymbol{X}_k + \boldsymbol{X}_{n-1} + \boldsymbol{\Xi}_n\right)\\
&\le \mathbb{E}\,\mathbb{E}_{n-2}\,\operatorname{tr}\exp\left(\sum\nolimits_{k=1}^{n-2}\boldsymbol{X}_k + \boldsymbol{\Xi}_{n-1} + \boldsymbol{\Xi}_n\right)\\
\cdots &\le \operatorname{tr}\exp\left(\sum\nolimits_{k=1}^{n}\boldsymbol{\Xi}_k\right).
\end{aligned}$$

We use the statistical independence of $\{\boldsymbol{X}_i\}$ to introduce the iterated expectation. At each step $m = 1, 2, 3, \ldots, n$, we invoke Corollary 1.5 with the fixed matrix $\boldsymbol{H}$ equal to

$$\boldsymbol{H}_m = \sum_{k=1}^{m-1}\boldsymbol{X}_k + \sum_{k=m+1}^{n}\boldsymbol{\Xi}_k.$$

This argument is legitimate because $\boldsymbol{H}_m$ is independent from $\boldsymbol{X}_m$.

The formulation (1.12) follows from (1.11) when we substitute the expression (1.6) for the matrix cgf and make some algebraic simplifications. ∎

1.3 Master bounds for sums of independent random matrices

We are now prepared to present some general results on the behavior of a sum of independent random matrices. In the next section, we derive some concrete matrix concentration inequalities using this approach.

1.3.1 The master inequalities

To obtain the main abstract results, we simply combine the Laplace transform bounds with the subadditivity of the matrix cgf.

Theorem 1.7 (Master bounds for a sum of independent random matrices). Consider a finite sequence $\{\boldsymbol{X}_k\}$ of independent, random, self-adjoint matrices of the same size. Then

$$\mathbb{E}\,\lambda_{\max}\left(\sum\nolimits_k \boldsymbol{X}_k\right) \le \inf_{\theta>0} \frac{1}{\theta} \log \operatorname{tr} \exp\left(\sum\nolimits_k \log \mathbb{E}\, \mathrm{e}^{\theta \boldsymbol{X}_k}\right); \tag{1.13}$$

$$\mathbb{E}\,\lambda_{\min}\left(\sum\nolimits_k \boldsymbol{X}_k\right) \ge \sup_{\theta<0} \frac{1}{\theta} \log \operatorname{tr} \exp\left(\sum\nolimits_k \log \mathbb{E}\, \mathrm{e}^{\theta \boldsymbol{X}_k}\right). \tag{1.14}$$

Furthermore, for all $t \in \mathbb{R}$,

$$\mathbb{P}\left\{\lambda_{\max}\left(\sum\nolimits_k \boldsymbol{X}_k\right) \ge t\right\} \le \inf_{\theta>0} \mathrm{e}^{-\theta t} \operatorname{tr} \exp\left(\sum\nolimits_k \log \mathbb{E}\, \mathrm{e}^{\theta \boldsymbol{X}_k}\right); \tag{1.15}$$

$$\mathbb{P}\left\{\lambda_{\min}\left(\sum\nolimits_k \boldsymbol{X}_k\right) \le t\right\} \le \inf_{\theta<0} \mathrm{e}^{-\theta t} \operatorname{tr} \exp\left(\sum\nolimits_k \log \mathbb{E}\, \mathrm{e}^{\theta \boldsymbol{X}_k}\right). \tag{1.16}$$

Proof. Substitute the subadditivity rule for matrix cgfs, Lemma 1.6, into the two matrix Laplace transform results, Proposition 1.1 and Proposition 1.2. ∎

1.3.2 Additional tools

To use Theorem 1.7, we need semidefinite bounds on the matrix cgf that reflect structural properties of the random matrices that appear in the sum. To implement this program, we need several basic facts from matrix analysis.

Fact 1.8 (Trace exponential is monotone). If $\boldsymbol{A} \preccurlyeq \boldsymbol{H}$, then $\operatorname{tr}\exp(\boldsymbol{A}) \le \operatorname{tr}\exp(\boldsymbol{H})$. ▪

Fact 1.9 (Logarithm is operator monotone). If $\boldsymbol{A} \preccurlyeq \boldsymbol{H}$, then $\log \boldsymbol{A} \preccurlyeq \log \boldsymbol{H}$. ▪

See [Tro15, Chap. 8] for the proofs of these results.

As a consequence of Fact 1.8, it suffices to produce semidefinite upper bounds for the matrix cgfs that appear in the formulas of Theorem 1.7. As a consequence of Fact 1.9, we can obtain a semidefinite upper bound for the matrix cgf from a semidefinite upper bound for the matrix mgf. We will see these ideas in action in the next section.

1.4 Example: Matrix Bernstein

We continue with the matrix Bernstein inequality, the matrix concentration result that has found the widest application. This result concerns a sum of independent zero-mean random matrices that are subject to a uniform norm bound.

1.4.1 Bernstein cgf bound

The first step in using Theorem 1.7 is to develop an estimate for the cgf of a bounded, zero-mean random matrix. This argument closely follows the analog argument in the scalar setting.

Lemma 1.10 (Bernstein cgf). Suppose that $\boldsymbol{X}$ is a random self-adjoint matrix that satisfies

$$\mathbb{E}\,\boldsymbol{X} = \mathbf{0} \quad \text{and} \quad \|\boldsymbol{X}\| \le 1.$$

Then

$$\log \mathbb{E}\, \mathrm{e}^{\theta \boldsymbol{X}} \preccurlyeq \frac{\theta^2/2}{1 - |\theta|/3} \cdot \mathbb{E}\,\boldsymbol{X}^2.$$

Proof. Suppose that $x \in [-1, +1]$. Using the Taylor series expansion of the exponential,

$$\begin{aligned} \mathrm{e}^{\theta x} &= 1 + \theta x + \sum\nolimits_{p=2}^{\infty} \frac{\theta^p}{p!} x^p \\ &\le 1 + \theta x + \left(\sum\nolimits_{p=2}^{\infty} \frac{|\theta|^p}{2 \cdot 3^{p-2}} \right) \cdot x^2 \\ &= 1 + \theta x + \frac{\theta^2/2}{1 - |\theta|/3} \cdot x^2. \end{aligned}$$

Since each eigenvalue of $\boldsymbol{X}$ lies in the interval $[-1, +1]$, we can apply this inequality to each eigenvalue of $\boldsymbol{X}$ to obtain

$$\mathrm{e}^{\theta \boldsymbol{X}} \preccurlyeq \mathbf{I} + \theta \boldsymbol{X} + \frac{\theta^2/2}{1 - |\theta|/3} \cdot \boldsymbol{X}^2.$$

Take the expectation:

$$\mathbb{E}\, \mathrm{e}^{\theta \boldsymbol{X}} \preccurlyeq \mathbf{I} + \frac{\theta^2/2}{1 - |\theta|/3} \cdot \mathbb{E}\,\boldsymbol{X}^2.$$

Invoke Fact 1.9:

$$\log \mathbb{E}\, \mathrm{e}^{\theta \boldsymbol{X}} \preccurlyeq \log \left(\mathbf{I} + \frac{\theta^2/2}{1 - |\theta|/3} \cdot \mathbb{E}\,\boldsymbol{X}^2 \right) \preccurlyeq \frac{\theta^2/2}{1 - |\theta|/3} \cdot \mathbb{E}\,\boldsymbol{X}^2.$$

The last relation follows when we apply the numerical inequality $\log(1 + x) \le x$, valid for $x > -1$, to each eigenvalue. ∎

1.4.2 The matrix Bernstein inequality

Combining the master tail bound, Theorem 1.7, with the cgf bound, Lemma 1.10, we arrive at the matrix Bernstein inequality.

Theorem 1.11 (Matrix Bernstein). Consider a statistically independent sequence $\{\boldsymbol{X}_k : 1 \le k \le n\}$ of random matrices with dimension d. Suppose that

$$\mathbb{E}\,\boldsymbol{X}_k = \mathbf{0} \quad \text{and} \quad \|\boldsymbol{X}_k\| \le B \quad \text{for each index } k.$$

Introduce the sum of the random matrices:

$$\boldsymbol{Y} = \sum\nolimits_{k=1}^{n} \boldsymbol{X}_k.$$

Define the matrix variance proxy:

$$\sigma^2 = \|\mathbb{E}\,\boldsymbol{Y}^2\| = \left\|\sum\nolimits_{k=1}^{n} \mathbb{E}\,\boldsymbol{X}_k^2\right\|.$$

Then, for all $t \geq 0$,

$$\mathbb{P}\{\|\boldsymbol{Y}\| \geq t\} \leq 2d \cdot \exp\left(\frac{-t^2/2}{\sigma^2 + Bt/3}\right).$$

Furthermore,

$$\mathbb{E}\,\|\boldsymbol{Y}\| \leq \sqrt{2\sigma^2 \log(2d)} + \frac{1}{3}B\log(2d).$$

Proof. First, rescale so that $B = 1$. The general form of the result follows from homogeneity arguments. The Bernstein cgf bound, Lemma 1.10, implies that

$$\log \mathbb{E}\,\mathrm{e}^{\theta \boldsymbol{X}_k} \preccurlyeq g(\theta)(\mathbb{E}\,\boldsymbol{X}_k^2) \quad \text{where} \quad g(\theta) = \frac{\theta^2/2}{1 - |\theta|/3}.$$

Note that $g(\theta) \geq 0$ for all $\theta \in \mathbb{R}$.

Substitute these cgf bounds into the master inequality (1.15) to obtain

$$\begin{aligned}
\mathbb{P}\{\lambda_{\max}(\boldsymbol{Y}) \geq t\} &\leq \inf_{\theta>0} \mathrm{e}^{-\theta t}\ \operatorname{tr}\exp\left(g(\theta)\sum\nolimits_k \mathbb{E}\,\boldsymbol{X}_k^2\right) \\
&\leq d \inf_{\theta>0} \mathrm{e}^{-\theta t}\ \lambda_{\max}\left(\exp\left(g(\theta)(\mathbb{E}\,\boldsymbol{Y}^2)\right)\right) \\
&= d \inf_{\theta>0} \mathrm{e}^{-\theta t}\ \exp\left(g(\theta)\,\sigma^2\right).
\end{aligned}$$

The first inequality depends on Fact 1.8. Afterward, we bound the trace by the dimension times the maximum eigenvalue. Next, we invoke the spectral mapping theorem and the fact that $g(\theta) > 0$ to draw the maximum eigenvalue inside the exponential. Identify the variance proxy σ^2 by noting that the maximum eigenvalue of the psd matrix $\mathbb{E}\,\boldsymbol{Y}^2$ coincides with its spectral norm.

Finally, we make the clever choice $\theta = t/(\sigma^2 + t/3)$ to see that

$$\mathbb{P}\{\lambda_{\max}(\boldsymbol{Y}) \geq t\} \leq d \cdot \exp\left(\frac{-t^2/2}{\sigma^2 + t/3}\right).$$

An identical argument yields a corresponding probability bound for the minimum eigenvalue of $\boldsymbol{Y}$. Combine the two results with the union bound to arrive at the stated probability bound for the spectral norm. ∎

Exercise 1.1 Use the master inequalities (1.13) and (1.14) to establish the expectation bound that appears in Theorem 1.11.

1.5 Example: Matrix Chernoff

As a second example, we develop bounds for the extreme eigenvalues of an independent sum of bounded, psd matrices.

1.5.1 Chernoff cgf bound

The matrix Chernoff inequality is based on the following cgf bound. It is a matrix version of a scalar argument.

Lemma 1.12 (Chernoff cgf). Suppose that $\boldsymbol{X}$ is a random self-adjoint matrix that satisfies

$$\mathbf{0} \preccurlyeq \boldsymbol{X} \preccurlyeq \mathbf{I}.$$

Then

$$\log \mathbb{E}\, \mathrm{e}^{\theta \boldsymbol{X}} \preccurlyeq (\mathrm{e}^{\theta} - 1)(\mathbb{E}\, \boldsymbol{X}) \quad \text{for } \theta \in \mathbb{R}.$$

This result is based on a classic computation for real random variables. The matrix extension first appeared in the proof of [AW02, Thm. 19]. See also [Tro12, Lem. 5.8].

Proof. The function $x \mapsto \mathrm{e}^{\theta x}$ is convex, so the graph lies below the chord connecting two points. In particular,

$$\mathrm{e}^{\theta x} \le 1 + (\mathrm{e}^{\theta} - 1)\, x \quad \text{for } x \in [0, 1].$$

The eigenvalues of $\boldsymbol{X}$ lie in the interval $[0, 1]$, so

$$\mathrm{e}^{\theta \boldsymbol{X}} \preccurlyeq \mathbf{I} + (\mathrm{e}^{\theta} - 1)\, \boldsymbol{X}.$$

Take the expectation:

$$\mathbb{E}\, \mathrm{e}^{\theta \boldsymbol{X}} \preccurlyeq \mathbf{I} + (\mathrm{e}^{\theta} - 1)(\mathbb{E}\, \boldsymbol{X}).$$

Arguing as in the proof of Lemma 1.10,

$$\log \mathbb{E}\, \mathrm{e}^{\theta \boldsymbol{X}} \preccurlyeq (\mathrm{e}^{\theta} - 1)(\mathbb{E}\, \boldsymbol{X}).$$

We have used Fact 1.9 and the numerical inequality $\log(1 + x) \le x$. ∎

1.5.2 Matrix Chernoff inequalities

Combining the master tail bound, Theorem 1.7, with the cgf bound, Lemma 1.12, we arrive at the matrix Chernoff inequalities.

Theorem 1.13 (Matrix Chernoff). Consider a statistically independent sequence $\{\boldsymbol{X}_k : 1 \le k \le n\}$ of random matrices with dimension d. Suppose that

$$\mathbf{0} \preccurlyeq \boldsymbol{X}_k \preccurlyeq B\, \mathbf{I} \quad \text{for each index } k.$$

Introduce the sum of the random matrices:

$$\boldsymbol{Y} = \sum\nolimits_{k=1}^{n} \boldsymbol{X}_k.$$

Define the lower and upper eigenvalues of the expectation:

$$\mu_{\min} = \lambda_{\min}(\mathbb{E}\, \boldsymbol{Y}) \quad \text{and} \quad \mu_{\max} = \lambda_{\max}(\mathbb{E}\, \boldsymbol{Y}).$$

Then

$$\mathbb{P}\{\lambda_{\min}(\boldsymbol{Y}) \le (1-\delta)\,\mu_{\min}\} \le d \cdot \left(\frac{\mathrm{e}^{-\delta}}{(1-\delta)^{1-\delta}}\right)^{\mu_{\min}/B} \quad \text{for } 0 < \delta \le 1;$$

$$\mathbb{P}\{\lambda_{\max}(\boldsymbol{Y}) \ge (1+\delta)\,\mu_{\max}\} \le d \cdot \left(\frac{\mathrm{e}^{\delta}}{(1+\delta)^{1+\delta}}\right)^{\mu_{\max}/B} \quad \text{for } \delta > 0.$$

Proof of Theorem 1.13, maximum eigenvalue bound. We begin with the tail bound for the maximum eigenvalue $\lambda_{\max}(\boldsymbol{Y})$. By a scaling argument, we may assume that $B = 1$. The Chernoff cgf bound, Lemma 1.12, implies that

$$\log \mathbb{E}\,\mathrm{e}^{\theta \boldsymbol{X}_k} \preccurlyeq g(\theta)(\mathbb{E}\,\boldsymbol{X}_k) \quad \text{where} \quad g(\theta) = \mathrm{e}^{\theta} - 1.$$

Note that $g(\theta) > 0$ for $\theta > 0$.

Using Fact 1.8, we substitute these cgf bounds into the master inequality (1.15) to reach

$$\begin{aligned}
\mathbb{P}\{\lambda_{\max}(\boldsymbol{Y}) \ge t\} &\le \inf_{\theta>0} \mathrm{e}^{-\theta t}\ \operatorname{tr}\exp\left(g(\theta)\sum\nolimits_k \mathbb{E}\,\boldsymbol{X}_k\right) \\
&\le \inf_{\theta>0} \mathrm{e}^{-\theta t}\ d\,\lambda_{\max}\big(\exp(g(\theta)(\mathbb{E}\,\boldsymbol{Y}))\big) \\
&= d \inf_{\theta>0} \mathrm{e}^{-\theta t}\ \exp\big(g(\theta)\,\lambda_{\max}(\mathbb{E}\,\boldsymbol{Y})\big) \\
&\le d \inf_{\theta>0} \mathrm{e}^{-\theta t}\ \exp\big(g(\theta)\,\mu_{\max}\big).
\end{aligned}$$

In the second line, we use the fact that the matrix exponential is pd to bound the trace by d times the maximum eigenvalue; we have also identified the sum as $\mathbb{E}\,\boldsymbol{Y}$. The third line follows from the spectral mapping theorem. Next, we use the fact that the maximum eigenvalue is a positive-homogeneous map, which depends on the observation that $g(\theta) > 0$ for $\theta > 0$. Finally, we identify the statistic $\mu_{\max}$.

To complete the proof, make the change of variables $t \mapsto (1+\delta)\,\mu_{\max}$. Then the infimum is achieved at $\theta = \log(1+\delta)$, which leads to the upper tail bound. ∎

The lower bounds follow from a related argument that is slightly more delicate.

Proof of Theorem 1.13, minimum eigenvalue bound. We now establish the bound for the minimum eigenvalue $\lambda_{\min}(\boldsymbol{Y})$. As before, rescale so that $B = 1$. The Chernoff cgf bound, Lemma 1.12, implies that

$$\log \mathbb{E}\,\mathrm{e}^{\theta \boldsymbol{X}_k} \preccurlyeq g(\theta)(\mathbb{E}\,\boldsymbol{X}_k) \quad \text{where} \quad g(\theta) = \mathrm{e}^{\theta} - 1.$$

Note that $g(\theta) < 0$ when $\theta < 0$.

Introduce these cgf bounds into the master inequality (1.16) to reach

$$\begin{aligned}
\mathbb{P}\{\lambda_{\min}(\boldsymbol{Y}) \le t\} &\le \inf_{\theta<0} \mathrm{e}^{-\theta t}\ \operatorname{tr}\exp\left(g(\theta)\sum\nolimits_k \mathbb{E}\,\boldsymbol{X}_k\right) \\
&\le \inf_{\theta<0} \mathrm{e}^{-\theta t}\ d\,\lambda_{\min}\big(\exp(g(\theta)(\mathbb{E}\,\boldsymbol{Y}))\big) \\
&= \inf_{\theta<0} \mathrm{e}^{-\theta t}\ d\,\exp\big(g(\theta)\,\lambda_{\min}(\mathbb{E}\,\boldsymbol{Y})\big) \\
&\le d \inf_{\theta<0} \mathrm{e}^{-\theta t}\,\exp\big(g(\theta)\cdot\mu_{\min}\big).
\end{aligned}$$

The justifications here are similar to those in the previous argument. The only noteworthy point is that we must replace the maximum eigenvalue map with the minimum eigenvalue map because $g(\theta) < 0$ for $\theta < 0$.

Finally, we make the change of variables $t \mapsto (1-\delta)\,\mu_{\min}$. The infimum is attained at $\theta = \log(1-\delta)$, which yields the lower tail bound. ∎

Exercise 1.2 Derive the following consequences of Theorem 1.13. For $\delta \in (0, 1]$,

$$\mathbb{P}\{\lambda_{\min}(\boldsymbol{Y}) \le (1-\delta)\mu_{\min}\} \le d \cdot \mathrm{e}^{-\delta^2 \mu_{\min}/(2B)};$$
$$\mathbb{P}\{\lambda_{\max}(\boldsymbol{Y}) \ge (1+\delta)\mu_{\max}\} \le d \cdot \mathrm{e}^{-\delta^2 \mu_{\max}/(3B)}.$$

These simplifications are often more tractable in practice.

1.6 The rectangular case

In these lectures, we will only be using matrix concentration for self-adjoint matrices. Nevertheless, it is important to be aware that concentration results for rectangular matrices follow as a formal consequence. This section outlines the approach.

1.6.1 The self-adjoint dilation

The *self-adjoint dilation* $\mathscr{H}(\boldsymbol{S})$ of a rectangular matrix $\boldsymbol{S} \in \mathbb{M}^{d_1 \times d_2}$ is the self-adjoint matrix

$$\mathscr{H}(\boldsymbol{S}) := \begin{bmatrix} \mathbf{0} & \boldsymbol{S} \\ \boldsymbol{S}^* & \mathbf{0} \end{bmatrix} \in \mathbb{H}_{d_1+d_2}. \tag{1.17}$$

Note that the map $\mathscr{H}$ is real-linear. By direct calculation,

$$\mathscr{H}(\boldsymbol{S})^2 = \begin{bmatrix} \boldsymbol{S}\boldsymbol{S}^* & \mathbf{0} \\ \mathbf{0} & \boldsymbol{S}^*\boldsymbol{S} \end{bmatrix}. \tag{1.18}$$

We also have the spectral identity

$$\lambda_{\max}(\mathscr{H}(\boldsymbol{S})) = \|\mathscr{H}(\boldsymbol{S})\| = \|\boldsymbol{S}\|. \tag{1.19}$$

This point follows from some linear algebraic considerations.

1.6.2 Rectangular matrix Bernstein

Using the device of the self-adjoint dilation, we can develop a version of the matrix Bernstein inequality for rectangular matrices.

Corollary 1.14 (Rectangular matrix Bernstein). Consider a statistically independent sequence $\{\boldsymbol{S}_k : 1 \le k \le n\}$ of $d_1 \times d_2$ random matrices. Suppose that

$$\mathbb{E}\,\boldsymbol{S}_k = \mathbf{0} \quad \text{and} \quad \|\boldsymbol{S}_k\| \le B \quad \text{for each index } k.$$

Introduce the sum of the random matrices:

$$\boldsymbol{Z} = \sum\nolimits_{k=1}^{n} \boldsymbol{S}_k.$$

Define the matrix variance proxy:

$$\begin{aligned}\sigma^2 &= \max\{\|\mathbb{E}\,\boldsymbol{Z}\boldsymbol{Z}^*\|,\ \|\mathbb{E}\,\boldsymbol{Z}^*\boldsymbol{Z}\|\} \\ &= \max\left\{\left\|\sum\nolimits_k \mathbb{E}\,\boldsymbol{S}_k\boldsymbol{S}_k^*\right\|,\ \left\|\sum\nolimits_k \mathbb{E}\,\boldsymbol{S}_k^*\boldsymbol{S}_k\right\|\right\}.\end{aligned}$$

Then, for all $t \geq 0$,

$$\mathbb{P}\{\|\boldsymbol{Z}\| \geq t\} \leq (d_1 + d_2) \cdot \exp\left(\frac{-t^2/2}{\sigma^2 + Bt/3}\right).$$

Furthermore,

$$\mathbb{E}\,\|\boldsymbol{Z}\| \leq \sqrt{2\sigma^2 \log(d_1 + d_2)} + \frac{1}{3}B\log(d_1 + d_2).$$

Exercise 1.3 Establish Corollary 1.14 by applying Theorem 1.11 to the self-adjoint dilation $\mathscr{H}(\boldsymbol{Z})$, perhaps with larger constants. **Hint:** To obtain the sharp constants presented here, you need to use the maximum eigenvalue bound that appears inside the proof of Theorem 1.11.

Notes

The modern theory of matrix concentration begins with the matrix Laplace transform technique (Proposition 1.1) developed by Ahlswede & Winter [AW02] and refined by Oliveira [Oli10]. The author of these notes recognized [Tro11a; Tro12; Tro15] that Lieb's theorem allows us to develop a perfect analogy (Theorem 1.7) with the scalar concentration theory. This idea has had a profound impact on computational mathematics over the last decade. These lectures explore some of the most striking outcomes.

Matrix concentration inequalities have a long history. Early work in operator theory and Banach space geometry includes [Buc01; Lus86; LP91; PX97; Rud99; Tom74]. The monograph [Tro15] provides a more comprehensive account.

2. Matrix Approximation by Sampling

"Corncobs" Wikimedia Commons

Most of the text in this lecture is copied from my monograph [Tro15, Chap. 6].

In applied mathematics, we often need to approximate a complicated target object by a more structured object. In some situations, we can solve this problem using a beautiful probabilistic approach called *empirical approximation*. The basic idea is to construct a "simple" random object whose expectation equals the target. We obtain the approximation by averaging several independent copies of the simple random object. As the number of terms in this average increases, the approximation becomes more complex, but it represents the target more faithfully. We must quantify this tradeoff.

In particular, we often encounter problems where we need to approximate a matrix by a more structured matrix. For example, we may wish to find a sparse matrix that is close to a given matrix, or we may need to construct a low-rank matrix that is close to a given matrix. Empirical approximation provides one mechanism for obtaining these approximations. The matrix Bernstein inequality offers a natural tool for assessing the quality of the randomized approximation.

This lecture develops a general framework for empirical approximation of symmetric matrices along with an application in machine learning. The monograph [Tro15, Chap. 6] includes the extension to rectangular matrices and several other basic applications.

2.1 Matrix sampling estimators

Let $\boldsymbol{A}$ be a self-adjoint target matrix that we hope to approximate by a more structured matrix. To that end, suppose we can represent the target as a sum of "simple" matrices:

$$\boldsymbol{A} = \sum\nolimits_{i=1}^{N} \boldsymbol{A}_i. \tag{2.1}$$

The idea is to identify summands $\boldsymbol{A}_i$ with desirable properties (such as sparsity or low rank) that we want our approximation to inherit.

Along with the decomposition (2.1), we need to construct a set of sampling probabilities:

$$\sum_{i=1}^{N} p_i = 1 \quad \text{and} \quad p_i > 0 \quad \text{for } i = 1, \dots, N. \tag{2.2}$$

We want to ascribe larger probabilities to "more important" summands. Quantifying what "important" means is the most difficult aspect of randomized matrix approximation. Choosing the right sampling distribution for a specific problem requires insight and ingenuity. Nevertheless, we will see that the matrix Bernstein inequality gives a strong hint about which distributions lead to the most accurate approximations.

Given the data (2.1) and (2.2), we may construct a "simple" random matrix $\boldsymbol{R}$ by sampling:

$$\boldsymbol{R} = p_i^{-1} \boldsymbol{A}_i \quad \text{with probability } p_i. \tag{2.3}$$

This construction ensures that $\boldsymbol{R}$ is an unbiased estimator of the target: $\mathbb{E}\,\boldsymbol{R} = \boldsymbol{A}$. Even so, the random matrix $\boldsymbol{R}$ offers a poor approximation of the target $\boldsymbol{A}$ because it has a lot more structure. To improve the quality of the approximation, we average n independent copies of the random matrix $\boldsymbol{R}$. We obtain an estimator of the form

$$\bar{\boldsymbol{R}}_n = \frac{1}{n} \sum_{k=1}^{n} \boldsymbol{R}_k \quad \text{where each } \boldsymbol{R}_k \text{ is an independent copy of } \boldsymbol{R}.$$

By linearity of expectation, this estimator is also unbiased: $\mathbb{E}\,\bar{\boldsymbol{R}}_n = \boldsymbol{A}$. The approximation $\bar{\boldsymbol{R}}_n$ remains structured when the number n of terms in the approximation is small as compared with the number N of terms in the decomposition (2.1).

Our goal is to quantify the approximation error as a function of the complexity n of the approximation:

$$\mathbb{E}\,\|\bar{\boldsymbol{R}}_n - \boldsymbol{A}\| \le \operatorname{err}(n).$$

As a reminder, $\|\cdot\|$ denotes the spectral norm; i.e., the ℓ_2 operator norm. There is a tension between the total number n of terms in the approximation and the error $\operatorname{err}(n)$ the approximation incurs. In applications, it is essential to achieve the right balance.

2.1.1 An error estimate

We can obtain an error estimate for the approximation scheme described in Section 2.1 as an immediate corollary of the matrix Bernstein inequality.

Theorem 2.1 (Matrix approximation by random sampling). Let $\boldsymbol{A} \in \mathbb{H}_d$ be a fixed matrix. Construct a random matrix $\boldsymbol{R} \in \mathbb{H}_d$ that satisfies

$$\mathbb{E}\,\boldsymbol{R} = \boldsymbol{A} \quad \text{and} \quad \|\boldsymbol{R}\| \le B.$$

Compute the per-sample second moment:

$$m_2(\boldsymbol{R}) = \|\mathbb{E}\,\boldsymbol{R}^2\|. \tag{2.4}$$

Form the matrix sampling estimator

$$\bar{\boldsymbol{R}}_n = \frac{1}{n}\sum\nolimits_{k=1}^{n} \boldsymbol{R}_k \quad \text{where each } \boldsymbol{R}_k \text{ is an independent copy of } \boldsymbol{R}.$$

Then the estimator satisfies, for all $t \geq 0$,

$$\mathbb{P}\left\{\|\bar{\boldsymbol{R}}_n - \boldsymbol{A}\| \geq t\right\} \leq 2d \exp\left(\frac{-nt^2/2}{m_2(\boldsymbol{R}) + 2Bt/3}\right). \tag{2.5}$$

Furthermore,

$$\mathbb{E}\,\|\bar{\boldsymbol{R}}_n - \boldsymbol{A}\| \leq \sqrt{\frac{2m_2(\boldsymbol{R})\log(2d)}{n}} + \frac{2B\log(2d)}{3n}. \tag{2.6}$$

Proof. Since $\boldsymbol{R}$ is an unbiased estimator of the target matrix $\boldsymbol{A}$, we can write

$$\boldsymbol{Y} = \bar{\boldsymbol{R}}_n - \boldsymbol{A} = \frac{1}{n}\sum\nolimits_{k=1}^{n}(\boldsymbol{R}_k - \mathbb{E}\,\boldsymbol{R}) = \sum\nolimits_{k=1}^{n}\boldsymbol{X}_k.$$

We have defined the summands $\boldsymbol{X}_k = n^{-1}(\boldsymbol{R}_k - \mathbb{E}\,\boldsymbol{R})$. These random matrices form an independent and identically distributed family, and each $\boldsymbol{X}_k$ has mean zero.

Now, each of the summands is subject to an upper bound:

$$\|\boldsymbol{X}_k\| \leq \frac{1}{n}\left(\|\boldsymbol{R}_k\| + \|\mathbb{E}\,\boldsymbol{R}\|\right) \leq \frac{1}{n}\left(\|\boldsymbol{R}_k\| + \mathbb{E}\,\|\boldsymbol{R}\|\right) \leq \frac{2B}{n}.$$

The first relation is the triangle inequality; the second is Jensen's inequality. The last estimate follows from our assumption that $\|\boldsymbol{R}\| \leq B$.

To control the matrix variance, first note that

$$\left\|\sum\nolimits_{k=1}^{n}\mathbb{E}\,\boldsymbol{X}_k^2\right\| = n\cdot\|\mathbb{E}\,\boldsymbol{X}_1^2\|.$$

The identity holds because the summands $\boldsymbol{X}_k$ are identically distributed. We may calculate that

$$\boldsymbol{0} \preccurlyeq \boldsymbol{X}_1^2 = n^{-2}\,\mathbb{E}(\boldsymbol{R} - \mathbb{E}\,\boldsymbol{R})^2 = n^{-2}\left[\,\mathbb{E}\,\boldsymbol{R}^2 - (\mathbb{E}\,\boldsymbol{R})^2\right] \preccurlyeq n^{-2}\,\mathbb{E}\,\boldsymbol{R}^2.$$

The first relation holds because the expectation of the random psd matrix $\boldsymbol{X}_1^2$ is psd. The first identity follows from the definition of $\boldsymbol{X}_1$ and the fact that $\boldsymbol{R}_1$ has the same distribution as $\boldsymbol{R}$. The second identity is a direct calculation. The last relation holds because $(\mathbb{E}\,\boldsymbol{R})^2$ is psd. In summary,

$$\left\|\sum\nolimits_{k=1}^{n}\mathbb{E}\,\boldsymbol{X}_k^2\right\| \leq \frac{1}{n}\|\mathbb{E}\,\boldsymbol{R}^2\| = \frac{m_2(\boldsymbol{R})}{n}.$$

The last line follows from the definition (2.4) of $m_2(\boldsymbol{R})$.

We are prepared to apply the matrix Bernstein inequality to the random matrix $\boldsymbol{Y}$. This act delivers the stated results. ∎

2.1.2 Discussion

One of the most common applications of the matrix Bernstein inequality is to analyze empirical matrix approximations. As a consequence, Corollary 2.1 is one of the most useful forms of the matrix Bernstein inequality. Let us discuss some of the important aspects of this result.

Understanding the bound on the approximation error

First, let us examine how many samples n suffice to bring the approximation error bound in Corollary 2.1 below a specified positive tolerance ε. Examining inequality (2.6), we find that

$$n \geq \frac{2m_2(\boldsymbol{R})\log(2d)}{\varepsilon^2} + \frac{2B\log(2d)}{3\varepsilon} \quad \text{implies} \quad \mathbb{E}\,\|\bar{\boldsymbol{R}}_n - \boldsymbol{A}\| \leq 2\varepsilon. \tag{2.7}$$

Roughly, the number n of samples should be on the scale of the maximum of the per-sample second moment $m_2(\boldsymbol{R})$ and the uniform upper bound B.

The bound (2.7) also reveals an unfortunate aspect of empirical matrix approximation. To make the tolerance ε small, the number n of samples must increase in proportion to ε^{-2}. In other words, it takes many samples to achieve a highly accurate approximation. We cannot avoid this phenomenon if we construct an approximation using an empirical average, because it is ultimately a consequence of the central limit theorem.

On a more positive note, it is quite valuable that the error bound (2.5) involves the spectral norm. This type of estimate simultaneously controls the error in every linear function of the approximation:

$$\|\bar{\boldsymbol{R}}_n - \boldsymbol{A}\| \leq \varepsilon \quad \text{implies} \quad |\operatorname{tr}(\bar{\boldsymbol{R}}_n\boldsymbol{C}) - \operatorname{tr}(\boldsymbol{A}\boldsymbol{C})| \leq \varepsilon \quad \text{for } \|\boldsymbol{C}\|_1 \leq 1.$$

We have written $\|\cdot\|_1$ for the Schatten 1-norm. These bounds also control the error in each eigenvalue $\lambda_j(\bar{\boldsymbol{R}}_n)$ of the approximation:

$$\|\bar{\boldsymbol{R}}_n - \boldsymbol{A}\| \leq \varepsilon \quad \text{implies} \quad |\lambda_j(\bar{\boldsymbol{R}}_n) - \lambda_j(\boldsymbol{A})| \leq \varepsilon.$$

When there is a gap between two eigenvalues of $\boldsymbol{A}$, we can also obtain bounds for the discrepancy between the associated eigenvectors of $\bar{\boldsymbol{R}}_n$ and $\boldsymbol{A}$ using perturbation theory [Bha97, Chap. VII].

Constructing empirical estimates

To obtain an accurate structured approximation, we need to select the right set of simple constituent matrices, as well as the right choice of sampling probabilities. In practice, these choices demand considerable creativity.

Fortunately, the matrix sampling result, Theorem 2.1, offers us some guidance because it identifies two summary parameters that control the quality of an empirical approximation. Indeed, we want to select the random matrix $\boldsymbol{R}$ to ensure that the upper bound B and the per-sample second moment $m_2(\boldsymbol{R})$ are both as small as possible. Later, we will see that this insight gives us a mechanism for determining the right sampling probabilities for certain problems.

This observation also hints at the possibility of achieving a bias–variance tradeoff when approximating $\boldsymbol{A}$. Indeed, we might drop all of the "unimportant" terms in the representation (2.1), i.e., those whose sampling probabilities are small. Then we construct a random approximation $\boldsymbol{R}$ only for the "important" terms that remain. Properly executed, this process may decrease both the per-sample second moment $m_2(\boldsymbol{R})$ and the upper bound B. The idea is analogous with shrinkage in statistical estimation.

A general sampling model

Corollary 2.1 extends beyond the sampling model based on the finite expansion (2.1). Indeed, we can consider a general decomposition of the self-adjoint target matrix $\boldsymbol{A}$:

$$\boldsymbol{A} = \int_{\Omega} \boldsymbol{A}(\omega) \, \mathrm{d}\mu(\omega), \tag{2.8}$$

where μ is a probability measure on a sample space Ω. As before, the idea is to represent the target matrix $\boldsymbol{A}$ as an average of "simple" matrices $\boldsymbol{A}(\omega)$. The main difference is that the family of simple matrices may now be infinite. In this setting, we construct the random approximation $\boldsymbol{R}$ so that

$$\mathbb{P}\{\boldsymbol{R} \in \mathsf{E}\} = \mu\{\omega : \boldsymbol{A}(\omega) \in \mathsf{E}\} \quad \text{for each Borel subset } \mathsf{E} \subseteq \mathbb{H}_d$$

In particular, it follows that

$$\mathbb{E}\,\boldsymbol{R} = \boldsymbol{A} \quad \text{and} \quad \|\boldsymbol{R}\| \le \sup_{\omega \in \Omega} \|\boldsymbol{A}(\omega)\|.$$

In this lecture, we will see how this abstraction allows us to approximate kernel matrices for machine learning applications.

Suboptimality of sampling estimators

Another fundamental point about sampling estimators is that they are often suboptimal. In other words, the matrix sampling estimator may incur an error substantially worse than the error in the best structured approximation of the target matrix.

To see why, let us consider a simple form of low-rank approximation by random sampling. The method here does not have practical value, but it highlights the reason that sampling estimators usually do not achieve ideal results. Suppose that $\boldsymbol{A}$ is a trace-one psd matrix with the eigenvalue decomposition

$$\boldsymbol{A} = \sum\nolimits_{i=1}^{d} \lambda_i \boldsymbol{u}_i \boldsymbol{u}_i^* \quad \text{where} \quad \sum\nolimits_{i=1}^{d} \lambda_i = 1 \quad \text{and} \quad \lambda_i \ge 0.$$

Given the eigenvalue decomposition, we can construct a random rank-one approximation $\boldsymbol{R}$ of the form

$$\boldsymbol{R} = \boldsymbol{u}_i \boldsymbol{u}_i^* \quad \text{with probability} \quad \lambda_i.$$

Per Corollary 2.1, the error in the associated sampling estimator $\bar{\boldsymbol{R}}_n$ is a rank-n matrix that satisfies

$$\|\bar{\boldsymbol{R}}_n - \boldsymbol{A}\| \le \sqrt{\frac{2\log(2d)}{n}} + \frac{2\log(2d)}{n}$$

On the other hand, a best rank-n approximation of $\boldsymbol{A}$ takes the form $\boldsymbol{A}_n = \sum_{j=1}^{n} \lambda_j \boldsymbol{u}_j \boldsymbol{u}_j^*$, and it incurs error

$$\|\boldsymbol{A}_n - \boldsymbol{A}\| = \lambda_{n+1} \leq \frac{1}{n+1}.$$

The second relation is Markov's inequality, which provides an accurate estimate only when the singular values $\lambda_1, \dots, \lambda_{n+1}$ are comparable. Regardless, the sampling estimator always incurs a somewhat larger error, which only converges as $n^{-1/2}$. Furthermore, there are many matrices whose singular values decay quickly, so that $\lambda_{n+1} \ll (n+1)^{-1}$. In the latter situation, the error in the sampling estimator is potentially much worse than the optimal error.

2.2 Application: Random features

As a first application of empirical matrix approximation, let us discuss an idea from machine learning called *random features*. The approach is based on the continuous sampling model (2.8), but it depends on the same principles as the discrete approximations that we introduced in Section 2.1.

Random feature maps were proposed by Ali Rahimi and Ben Recht [RR07], and they have turned out to be useful in practice. The analysis in this section is due to David Lopez-Paz et al. [Lop+14].

2.2.1 Kernel matrices

Let $\mathscr{X}$ be a set. We think about the elements of the set $\mathscr{X}$ as (potential) observations that we would like to use to perform learning and inference tasks. Let us introduce a bounded measure K of similarity between pairs of points in the set:

$$K : \mathscr{X} \times \mathscr{X} \to [-1, +1].$$

The similarity measure K is often called a *kernel*. We assume that the kernel returns the value $+1$ when its arguments are identical, and it returns smaller values when its arguments are dissimilar. We also assume that the kernel is symmetric; that is, $K(\boldsymbol{x}, \boldsymbol{y}) = K(\boldsymbol{y}, \boldsymbol{x})$ for all arguments $\boldsymbol{x}, \boldsymbol{y} \in \mathscr{X}$.

A simple example of a kernel is the angular similarity between a pair of points in a Euclidean space:

$$K(\boldsymbol{x}, \boldsymbol{y}) = \frac{2}{\pi} \arcsin \frac{\langle \boldsymbol{x}, \boldsymbol{y} \rangle}{\|\boldsymbol{x}\| \|\boldsymbol{y}\|} = 1 - \frac{2}{\pi} \cdot \measuredangle(\boldsymbol{x}, \boldsymbol{y}) \quad \text{for } \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d. \tag{2.9}$$

We write $\measuredangle(\cdot, \cdot)$ for the planar angle between two vectors, measured in radians. As usual, we instate the convention that $0/0 = 0$. See Figure 2.1 for an illustration.

Suppose that $\boldsymbol{x}_1, \dots, \boldsymbol{x}_N \in \mathscr{X}$ are observations. The $N \times N$ kernel matrix $\boldsymbol{G} = [g_{ij}]$ tabulates the values of the kernel function for each pair of data points:

$$g_{ij} = K(\boldsymbol{x}_i, \boldsymbol{x}_j) \quad \text{for } i, j = 1, \dots, N.$$

We say that the kernel K is *positive definite* if the kernel matrix $\boldsymbol{G}$ is positive semidefinite for any choice of observations $\{\boldsymbol{x}_i\} \subset \mathscr{X}$. We will be concerned only with positive-definite kernels in this discussion. It may be helpful to think about the kernel matrix $\boldsymbol{G}$ as a generalization of the Gram matrix of a family of points in a Euclidean space.

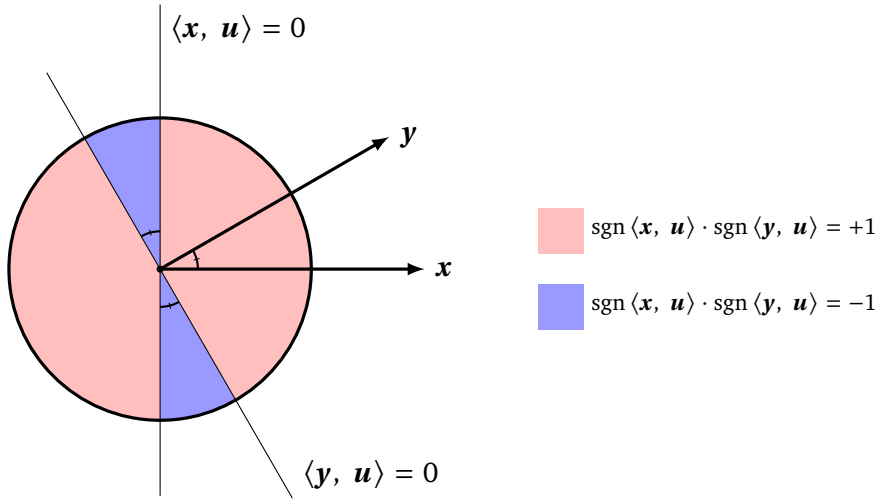


Figure 2.1: **The angular similarity between two vectors.** Let $\boldsymbol{x}$ and $\boldsymbol{y}$ be nonzero vectors in $\mathbb{R}^2$ with angle $\angle(\boldsymbol{x}, \boldsymbol{y})$. The light red region contains the directions $\boldsymbol{u}$ where the product $\operatorname{sgn}\langle \boldsymbol{x}, \boldsymbol{u}\rangle \cdot \operatorname{sgn}\langle \boldsymbol{y}, \boldsymbol{u}\rangle$ equals $+1$, and the dark blue region contains the directions $\boldsymbol{u}$ where the same product equals -1. The blue region subtends a total angle of $2\angle(\boldsymbol{x}, \boldsymbol{y})$, and the red region subtends a total angle of $2\pi - 2\angle(\boldsymbol{x}, \boldsymbol{y})$.

In the Euclidean setting, there are many statistical learning methods that only require the inner product between each pair of observations. These algorithms can be extended to the kernel setting by replacing each inner product with a kernel evaluation. As a consequence, kernel matrices can be used for classification, regression, and feature selection. In these applications, kernels are advantageous because they work outside the Euclidean domain, and they allow task-specific measures of similarity. This idea, sometimes called the *kernel trick*, is a major insight with wide applications [SS01].

A significant challenge for algorithms based on kernels is that the kernel matrix is big. Indeed, $\boldsymbol{G}$ contains $\Theta(N^2)$ entries, where N is the number of data points. Furthermore, the cost of constructing the kernel matrix is often $\Theta(dN^2)$ where d is the number of parameters required to specify a point in the universe $\mathscr{X}$.

Nevertheless, there is an opportunity. Large data sets tend to be redundant, so the kernel matrix also tends to be redundant. This event manifests in the kernel matrix being well-approximated by a low-rank matrix. As a consequence, we may try to replace the kernel matrix by a low-rank proxy. For some similarity measures, we can accomplish this task using empirical approximation.

2.2.2 Random features and low-rank approximation of the kernel matrix

In certain cases, a positive-definite kernel can be written as an expectation (2.8), and we can take advantage of this representation to construct an empirical approximation of the kernel matrix. Let us begin with the general construction, and then we will present a few examples in Section 2.2.3.

Let $\mathscr{W}$ be a sample space equipped with a sigma-algebra and a probability measure

μ. Introduce a bounded *feature map*:

$$\psi : \mathscr{X} \times \mathscr{W} \to [-b, +b] \quad \text{where } b > 0.$$

Consider a random variable $\boldsymbol{w}$ taking values in $\mathscr{W}$ and distributed according to the measure μ. We assume that this random variable satisfies the *reproducing property*

$$K(\boldsymbol{x}, \boldsymbol{y}) = \mathbb{E}_{\boldsymbol{w}} \left[\psi(\boldsymbol{x}; \boldsymbol{w}) \cdot \psi(\boldsymbol{y}; \boldsymbol{w}) \right] \quad \text{for all } \boldsymbol{x}, \boldsymbol{y} \in \mathscr{X}. \tag{2.10}$$

The pair $(\psi, \boldsymbol{w})$ is called a *random feature map* for the kernel K. As we will see, this hypothesis will lead to an instance of the expectation model (2.8) for the kernel matrix of an arbitrary dataset.

We want to approximate the kernel matrix associated with a set $\{\boldsymbol{x}_1, \dots, \boldsymbol{x}_N\} \subset \mathscr{X}$ of observations. To do so, we draw a random vector $\boldsymbol{w} \in \mathscr{W}$ distributed according to μ. Form a random vector $\boldsymbol{z} \in \mathbb{R}^N$ by applying the feature map to each data point with the *same* choice of the random vector $\boldsymbol{w}$. That is,

$$\boldsymbol{z} = \begin{bmatrix} z_1 \\ \vdots \\ z_N \end{bmatrix} = \begin{bmatrix} \psi(\boldsymbol{x}_1; \boldsymbol{w}) \\ \vdots \\ \psi(\boldsymbol{x}_N; \boldsymbol{w}) \end{bmatrix}.$$

The vector $\boldsymbol{z} \in \mathbb{R}^N$ is sometimes called a *random feature*; it should be regarded as a summary of the entire dataset. By the reproducing property (2.10) for the random feature map, for each pair (i, j) of indices,

$$g_{ij} = K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \mathbb{E}_{\boldsymbol{w}} \left[\psi(\boldsymbol{x}_i; \boldsymbol{w}) \cdot \psi(\boldsymbol{x}_j; \boldsymbol{w}) \right] = \mathbb{E}_{\boldsymbol{w}} \left[z_i \cdot z_j \right].$$

In other words, the feature map gives us an unbiased estimator for each entry of the kernel matrix.

We can write this relation in matrix form as

$$\boldsymbol{G} = \mathbb{E}[\boldsymbol{z}\boldsymbol{z}^*].$$

The random matrix $\boldsymbol{R} = \boldsymbol{z}\boldsymbol{z}^*$ is an unbiased rank-one estimator for the kernel matrix $\boldsymbol{G}$. This is an instantiation of the model (2.8)! Note that this representation demonstrates that random feature maps, as defined here, only exist for positive-definite kernels. (But we can construct random feature maps for some other kinds of kernels using related approaches.)

We can construct a better empirical approximation of the kernel matrix $\boldsymbol{G}$ by averaging realizations of the simple estimator $\boldsymbol{R}$:

$$\bar{\boldsymbol{R}}_n = \frac{1}{n} \sum\nolimits_{k=1}^{n} \boldsymbol{R}_k \quad \text{where } \boldsymbol{R}_k \text{ is an independent copy of } \boldsymbol{R}. \tag{2.11}$$

In other words, we are using n independent random features $\boldsymbol{z}_1, \dots, \boldsymbol{z}_n$ to approximate the kernel matrix.

The cost of computing a single random feature is typically $\Theta(dN)$, where d is the number of parameters required to specify a point in the universe $\mathscr{X}$. Therefore, the cost of computing n random features is $\Theta(dnN)$. When $n \ll N$, the cost of obtaining the random feature approximation $\bar{\boldsymbol{R}}_n$ is substantially smaller than the cost of computing the full kernel matrix. The question is how many random features n we needed before our estimator is accurate.

2.2.3 Examples of random feature maps

Before we continue with the analysis, let us describe some random feature maps. This discussion is tangential to our theme of matrix concentration, but it is valuable to understand why random feature maps exist.

The angular similarity kernel

First, let us consider the angular similarity (2.9) defined on $\mathbb{R}^d$. We can construct a random feature map using a classic result from plane geometry. If we draw $\boldsymbol{w}$ uniformly from the unit sphere $\mathbb{S}^{d-1} \subset \mathbb{R}^d$, then

$$\begin{aligned} K(\boldsymbol{x}; \boldsymbol{y}) &= 1 - \frac{2}{\pi} \cdot \measuredangle(\boldsymbol{x}, \boldsymbol{y}) \\ &= \mathbb{E}_{\boldsymbol{w}} \left[\operatorname{sgn} \langle \boldsymbol{x}, \boldsymbol{w} \rangle \cdot \operatorname{sgn} \langle \boldsymbol{y}, \boldsymbol{w} \rangle \right] \quad \text{for all } \boldsymbol{x}, \boldsymbol{y} \in \mathscr{X}. \end{aligned} \tag{2.12}$$

The easy proof of this relation should be visible from the diagram in Figure 2.1. In light of the formula (2.12), we set $\mathscr{W} = \mathbb{S}^{d-1}$ with the uniform measure, and we define the feature map

$$\psi(\boldsymbol{x}; \boldsymbol{w}) = \operatorname{sgn} \langle \boldsymbol{x}, \boldsymbol{w} \rangle.$$

The reproducing property (2.10) follows immediately from (2.12). Therefore, the pair $(\psi, \boldsymbol{w})$ is a random feature map for the angular similarity kernel.

The paper [KK12] explains how to compute random features for more general inner-product kernels using a classic theorem of Schönberg.

Translation-invariant kernels

Next, let us describe an important class of kernels that can be expressed using random feature maps. A kernel on $\mathbb{R}^d$ is *translation invariant* if there is a function $\varphi : \mathbb{R}^d \to \mathbb{R}$ for which

$$K(\boldsymbol{x}, \boldsymbol{y}) = \varphi(\boldsymbol{x} - \boldsymbol{y}) \quad \text{for all } \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d.$$

Bôchner's theorem, a classical result from harmonic analysis, gives a representation for each continuous, positive-definite, translation-invariant kernel:

$$\begin{aligned} K(\boldsymbol{x}, \boldsymbol{y}) &= \varphi(\boldsymbol{x} - \boldsymbol{y}) \\ &= c \int_{\mathbb{R}^d} \mathrm{e}^{\mathrm{i}\langle \boldsymbol{x}, \boldsymbol{w} \rangle} \cdot \mathrm{e}^{-\mathrm{i}\langle \boldsymbol{y}, \boldsymbol{w} \rangle} \, \mathrm{d}\mu(\boldsymbol{w}) \quad \text{for all } \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d. \end{aligned} \tag{2.13}$$

In this expression, c is a positive scale factor c, and μ is a probability measure on $\mathbb{R}^d$, and these objects depend only on the function φ. Conversely, for any probability measure μ, the formula (2.13) induces a continuous, positive-definite, translation-invariant kernel.

Bôchner's theorem (2.13) allows us to construct a (complex-valued) random feature map for the kernel K:

$$\psi_{\mathbb{C}}(\boldsymbol{x}; \boldsymbol{w}) = \sqrt{c}\, \mathrm{e}^{\mathrm{i}\langle \boldsymbol{x}, \boldsymbol{w} \rangle} \quad \text{where } \boldsymbol{w} \text{ has distribution } \mu \text{ on } \mathbb{R}^d.$$

This map satisfies a complex variant of the reproducing property (2.10):

$$K(\boldsymbol{x}, \boldsymbol{y}) = \mathbb{E}_{\boldsymbol{w}} \left[\psi_{\mathbb{C}}(\boldsymbol{x}; \boldsymbol{w}) \cdot \psi_{\mathbb{C}}(\boldsymbol{y}; \boldsymbol{w})^* \right] \quad \text{for all } \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d,$$

where we have written * for complex conjugation.

With a little more work, we can construct a real-valued random feature map. Recall that the kernel K is symmetric, so the complex exponentials in (2.13) can be written in terms of cosines. This observation leads to the random feature map

$$\psi(\boldsymbol{x}; \boldsymbol{w}, U) = \sqrt{2c}\cos\big(\langle \boldsymbol{x},\ \boldsymbol{w}\rangle + U\big) \quad \text{where} \quad \boldsymbol{w} \sim \mu \text{ and } U \sim \textsc{uniform}[0, 2\pi]. \tag{2.14}$$

To verify that $(\psi, (\boldsymbol{w}, U))$ reproduces the kernel K, as required by (2.10), we just make a short calculation using the angle-sum formula for the cosine.

We conclude this section with the most important example of a random feature map from the class we have just described. Consider the Gaussian radial basis function kernel:

$$K(\boldsymbol{x}, \boldsymbol{y}) = \mathrm{e}^{-\alpha\|\boldsymbol{x}-\boldsymbol{y}\|^2/2} \quad \text{for all } \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d.$$

The positive parameter α reflects how close two points must be before they are regarded as "similar." For the Gaussian kernel, Bôchner's Theorem (2.13) holds with the scaling factor $c = 1$ and the probability measure $\mu = \textsc{normal}(\mathbf{0}, \alpha\mathbf{I}_d)$. In summary, we define

$$\psi(\boldsymbol{x}; \boldsymbol{w}, U) = \sqrt{2}\cos\big(\langle \boldsymbol{x},\ \boldsymbol{w}\rangle + U\big) \quad \text{where} \quad \boldsymbol{w} \sim \textsc{normal}(\mathbf{0}, \alpha\mathbf{I}_d) \text{ and } U \sim \textsc{uniform}[0, 2\pi].$$

This random feature map reproduces the Gaussian radial basis function kernel.

2.2.4 Error bound for the random feature approximation

We will demonstrate that the approximation $\bar{\boldsymbol{R}}_n$ of the $N \times N$ kernel matrix $\boldsymbol{G}$ using n random features, constructed in (2.11), leads to an estimate of the form

$$\mathbb{E}\,\|\bar{\boldsymbol{R}}_n - \boldsymbol{G}\| \le \sqrt{\frac{2bN\|\boldsymbol{G}\|\log(2N)}{n}} + \frac{2bN\log(2N)}{3n}. \tag{2.15}$$

In this expression, b is the uniform bound on the magnitude of the feature map ψ. The short proof of (2.15) appears in Section 2.2.5.

To clarify what this result means, we introduce the *intrinsic dimension* of the $N \times N$ kernel matrix $\boldsymbol{G}$:

$$\operatorname{intdim}(\boldsymbol{G}) = \frac{\operatorname{tr}\boldsymbol{G}}{\|\boldsymbol{G}\|} = \frac{N}{\|\boldsymbol{G}\|}.$$

Note that $\operatorname{tr}\boldsymbol{G} = N$ because of the requirement that $K(\boldsymbol{x}, \boldsymbol{x}) = +1$ for all $\boldsymbol{x} \in \mathcal{X}$. The intrinsic dimension $\operatorname{intdim}(\boldsymbol{G})$ is a continuous measure of the number of energetic dimensions, and it is always bounded above by the algebraic rank of $\boldsymbol{G}$.

Now, assume that the number n of random features satisfies the bound

$$n \ge 2b\varepsilon^{-2} \cdot \operatorname{intdim}(\boldsymbol{G}) \cdot \log(2N),$$

In view of (2.15), the relative error in the empirical approximation of the kernel matrix satisfies

$$\frac{\mathbb{E}\,\|\bar{\boldsymbol{R}}_n - \boldsymbol{G}\|}{\|\boldsymbol{G}\|} \le \varepsilon + \varepsilon^{-2}.$$

We learn that the randomized approximation of the kernel matrix $\boldsymbol{G}$ using n random features can be accurate when n is proportional to the intrinsic dimension of $\boldsymbol{G}$, even if the intrinsic dimension is much smaller than the number of data points. That is, $n \approx \operatorname{intdim}(\boldsymbol{G}) \ll N$.

2.2.5 Analysis of the random feature approximation

The analysis of random features is based on Corollary 2.1. To apply this result, we need the per-sample second-moment $m_2(\boldsymbol{R})$ and the uniform upper bound B. Both are easy to come by.

First, observe that

$$\|\boldsymbol{R}\| = \|\boldsymbol{z}\boldsymbol{z}^*\| = \|\boldsymbol{z}\|^2 \leq bN$$

Recall that b is the uniform bound on the feature map ψ, and N is the number of components in the random feature vector $\boldsymbol{z}$.

Second, we calculate that

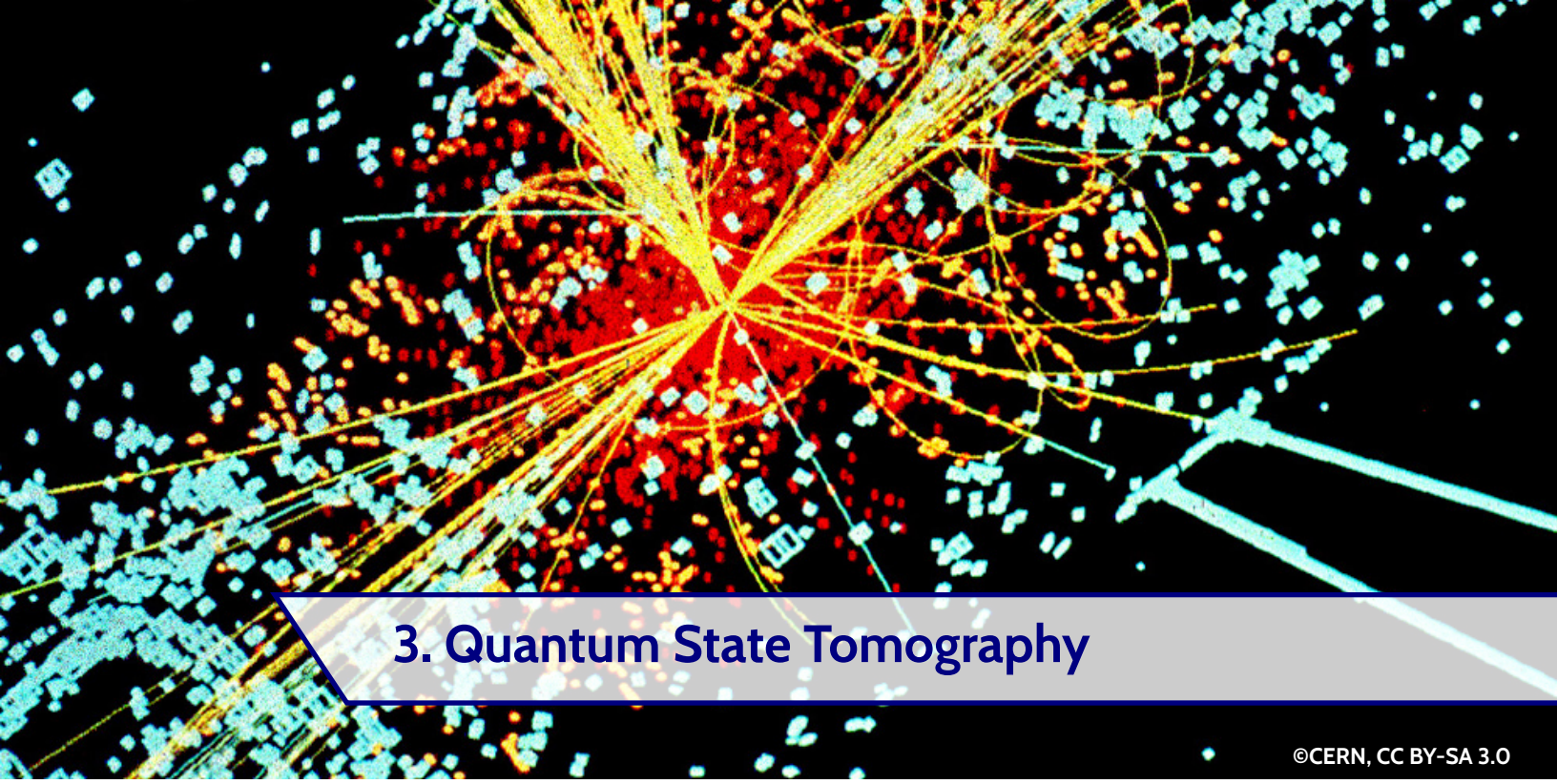
$$\mathbb{E}\,\boldsymbol{R}^2 = \mathbb{E}\left[\|\boldsymbol{z}\|^2\,\boldsymbol{z}\boldsymbol{z}^*\right] \preccurlyeq bN \cdot \mathbb{E}[\boldsymbol{z}\boldsymbol{z}^*] = bN \cdot \boldsymbol{G}.$$

Each random matrix $\boldsymbol{z}\boldsymbol{z}^*$ is positive semidefinite, so we can introduce the upper bound $\|\boldsymbol{z}\|^2 \leq bN$. The last identity holds because $\boldsymbol{R}$ is an unbiased estimator of the kernel matrix $\boldsymbol{G}$. It follows that

$$m_2(\boldsymbol{R}) = \|\mathbb{E}\,\boldsymbol{R}^2\| \leq bN \cdot \|\boldsymbol{G}\|.$$

This is our bound for the per-sample second moment.

Finally, we invoke Corollary 2.1 with parameters $B = bN$ and $m_2(\boldsymbol{R}) \leq bN\|\boldsymbol{G}\|$ to arrive at the estimate (2.15).



3. Quantum State Tomography

This lecture was written primarily by Richard Kueng, on the basis of our joint work [Guh+18]. Any errors that appear are the fault of the lecturer.

A core problem in quantum information science is to estimate the state of a quantum system from measurements (of multiple realizations) of the system. This problem is called *quantum tomography*. In quantum computing, the state is represented by a finite-dimensional matrix, so we can formulate the tomography problem as a question about matrix estimation.

This lecture considers a special class of quantum tomography problems that admit a particularly simple analysis based on the matrix Bernstein inequality (Theorem 2.1). A remarkable feature of this application is that random matrices arise as a consequence of quantum mechanics!

3.1 Postulates of quantum mechanics

Quantum mechanics is a *probabilistic theory*, contra Einstein's firm belief that "God does not play dice." In this lecture, we will restrict ourselves to finite-dimensional quantum mechanics, where the principles are clearest. The extension to infinite dimensions is conceptually straightforward, and it resembles the transition from matrix analysis to functional analysis. To begin, we will develop the fundamental axioms of quantum mechanics as a noncommutative extension of discrete probability theory.

3.1.1 Recapitulation: Discrete probability theory

Recall that $\langle \cdot, \cdot \rangle$ is the standard inner product on $\mathbb{R}^d$. We denote the vector of ones by $\mathbf{1} = (1, \dots, 1)^* \in \mathbb{R}^d$; this is the unit for the Hadamard product of vectors.

A discrete probability distribution on d points is fully characterized by a d-dimensional probability vector. A probability vector is just a nonnegative vector whose entries sum to one. The set Δ_d of all d-dimensional probability vectors is called the *probability simplex*:

$$\Delta_d = \{\boldsymbol{p} \in \mathbb{R}^d : \boldsymbol{p} \geq \mathbf{0} \text{ and } \langle \mathbf{1}, \boldsymbol{p} \rangle = 1\}.$$

The probability simplex is a compact, convex set of vectors. The extreme points $\boldsymbol{\delta}_i$ of the probability simplex are the nonrandom probability distributions. The barycenter $d^{-1}\mathbf{1}$ of the probability simplex is the uniform distribution.

Elementary events (singleton outcomes) are encoded by the d standard basis vectors $\boldsymbol{\delta}_1, \ldots, \boldsymbol{\delta}_d \in \mathbb{R}^d$. Thus, the probability rule is given by the inner product:

$$\mathbb{P}\{i \,|\, \boldsymbol{p}\} = \langle \boldsymbol{\delta}_i, \boldsymbol{p} \rangle = p_i \in [0, 1].$$

An event E is an element of the power set of $\{1, \ldots, d\}$. We may represent the event E by the binary indicator vector $\mathbb{1}_{\mathsf{E}} \in \{0, 1\}^n$. The probability rule remains the same:

$$\mathbb{P}\{\mathsf{E} \,|\, \boldsymbol{p}\} = \langle \mathbb{1}_{\mathsf{E}}, \boldsymbol{p} \rangle = \sum\nolimits_{i \in \mathsf{E}} p_i \in [0, 1].$$

This formalism extends to convex mixtures of events, which we call *generalized events*. The family of generalized events coincides with the standard cube:

$$\operatorname{conv}\{0, 1\}^d = \{\boldsymbol{h} \in \mathbb{R}^n : \mathbf{0} \leq \boldsymbol{h} \leq \mathbf{1}\} = \mathsf{Q}_d.$$

The probability of a generalized event $\boldsymbol{h} \in \mathsf{Q}_d$ is given by the inner product $\langle \boldsymbol{h}, \boldsymbol{p} \rangle$. We can therefore associate generalized events with the class of nonnegative random variables that are bounded by one, and the probability of a generalized event is the expectation of this random variable. In summary, generalized events are dual to probability distributions.

Next, we define the notion of a classical measurement.

Definition 3.1 (Classical measurement). A (classical) *measurement* $\{\boldsymbol{h}_{\lambda_1}, \ldots, \boldsymbol{h}_{\lambda_m}\} \subset \mathsf{Q}_d$ is a set of generalized events that forms a resolution of the vector of ones:

$$\mathbf{0} \leq \boldsymbol{h}_{\lambda_i} \leq \mathbf{1} \quad \text{and} \quad \sum\nolimits_{i=1}^m \boldsymbol{h}_{\lambda_i} = \mathbf{1}.$$

A measurement should be viewed as a complete set of (generalized) events.

$$\sum\nolimits_{i=1}^m \mathbb{P}\{\lambda_i \,|\, \boldsymbol{p}\} = \sum\nolimits_{i=1}^m \langle \boldsymbol{h}_{\lambda_i}, \boldsymbol{p} \rangle = \langle \mathbf{1}, \boldsymbol{p} \rangle = 1.$$

In other words, it is certain that one of the outcomes $\lambda_1, \ldots, \lambda_m$ will occur.

Example 3.1 (What is a classical measurement?). Suppose that you and I agree on a bet that involves two random variables: a fair coin toss and the roll of a die. We first toss the coin and subsequently roll the die. The rules for victory depend on the outcome of the coin toss:

1. If the coin comes up heads, then I win if the die produces an odd number $\{1, 3, 5\}$. Otherwise, you win.

2. If the coin comes up tails, then I win if the die produces a number in the set $\{1, 2, 3\}$. Otherwise, you win.

A generalized classical event allows us to absorb the randomness in the coin flip into a generalized event that is associated only with the outcome of rolling the die:

$$\boldsymbol{h}_{\text{I win}} = \frac{1}{2}(1, 0, 1, 0, 1, 0) + \frac{1}{2}(1, 1, 1, 0, 0, 0) = (1, 0.5, 1, 0, 0.5, 0);$$
$$\boldsymbol{h}_{\text{You win}} = \frac{1}{2}(0, 1, 0, 1, 0, 1) + \frac{1}{2}(0, 0, 0, 1, 1, 1) = (0, 0.5, 0, 1, 0.5, 1).$$

In other words, these two generalized events arise from marginalization over the first variable. The generalized events reflect our suspense about who will win the game once we roll the die.

The pair $\{\boldsymbol{h}_{\text{I win}}, \boldsymbol{h}_{\text{You win}}\}$ constitutes a classical measurement system. In this case, performing the measurement amounts to completing the game (by rolling the die) and recording the outcome.

An alternative perspective, that is more quantum in spirit, realizes the probability rule as a tensor product and "sums out" one of the components. In the matrix setting, the analogous operation is called a partial trace. ∎

Table 3.1 summarizes the basic concepts of classical discrete probability theory.

Concept	**Representation**	**Formula**
Probability density	Normalized, nonnegative $\boldsymbol{p} \in \mathbb{R}^d$	$\boldsymbol{p} \geq \mathbf{0}$ and $\langle \mathbf{1}, \boldsymbol{p} \rangle = 1$
Measurement	Resolution $\{\boldsymbol{h}_{\lambda_i}\}$ of the unit $\mathbf{1}$	$\boldsymbol{h}_{\lambda_i} \geq \mathbf{0}$ and $\sum_{i=1}^m \boldsymbol{h}_{\lambda_i} = \mathbf{1}$
Probability rule	Standard inner product	$\mathbb{P}\{\lambda_i \mid \boldsymbol{p}\} = \langle \boldsymbol{h}_{\lambda_i}, \boldsymbol{p} \rangle$

Table 3.1: **Axioms for classical probability theory.** The structure of discrete probability theory is captured by endowing $\mathbb{R}^d$ with the partial order $\geq$ and the identity element $\mathbf{1}$.

3.1.2 Noncommutative probability theory

The postulates of quantum mechanics arise naturally from a noncommutative extension of classical probability theory. We simply replace the triple $(\mathbb{R}^d, \geq, \mathbf{1})$ by the triple $(\mathbb{H}_d, \succcurlyeq, \mathbf{I})$. Recall that $\mathbb{H}_d = \mathbb{H}_d(\mathbb{C})$ is the space of self-adjoint $d \times d$ complex matrices, endowed with the trace inner product $\langle \boldsymbol{X}, \boldsymbol{Y} \rangle = \operatorname{tr}(\boldsymbol{XY})$, the semidefinite order $\succcurlyeq$, and the identity matrix $\mathbf{I}$.

In quantum mechanics, the analog of a probability density vector is a *(probability) density matrix*.

Definition 3.2 (Density matrix). The state of a d-dimensional quantum mechanical system is fully described by a *density matrix* $\boldsymbol{\rho} \in \mathbb{H}_d$, a self-adjoint matrix that satisfies

$$\boldsymbol{\rho} \succcurlyeq \mathbf{0} \quad \text{and} \quad \langle \mathbf{I}, \boldsymbol{\rho} \rangle = \operatorname{tr}(\boldsymbol{\rho}) = 1.$$

Density matrices are often called *states*.

Introduce the family $\mathcal{S}(\mathbb{H}_d)$ of all d-dimensional density matrices:

$$\mathcal{S}(\mathbb{H}_d) = \{\boldsymbol{X} \in \mathbb{H}_d : \boldsymbol{X} \succcurlyeq \mathbf{0} \text{ and } \langle \mathbf{I}, \boldsymbol{X} \rangle = 1\}$$

Like the probability simplex, the set $\mathcal{S}(\mathbb{H}_d)$ of density matrices is compact and convex.

In parallel to a classical measurement system, we may now define a quantum measurement system.

Definition 3.3 (Quantum measurement). A (quantum) *measurement* is a collection $\{\boldsymbol{H}_{\lambda_i} : 1 \le i \le m\}$ of psd matrices that forms a resolution of the identity matrix:

$$\mathbf{0} \preccurlyeq \boldsymbol{H}_{\lambda_i} \preccurlyeq \mathbf{I} \quad \text{and} \quad \sum\nolimits_{i=1}^{m} \boldsymbol{H}_{\lambda_i} = \mathbf{I}.$$

When a measurement $\{\boldsymbol{H}_{\lambda_i} : 1 \le i \le m\}$ is performed on a quantum mechanical system with density matrix $\boldsymbol{\rho}$, two things happen.

1. **Born's rule:** We obtain a random measurement outcome λ_i that follows the probability distribution

$$\mathbb{P}\{\lambda_i \mid \boldsymbol{\rho}\} = \langle \boldsymbol{H}_{\lambda_i}, \boldsymbol{\rho} \rangle = \operatorname{tr}(\boldsymbol{H}_{\lambda_i} \boldsymbol{\rho}). \tag{3.1}$$

2. **Collapse of wavefunction:** The quantum system ceases to exist.

There is some philosophical debate about this model, but experimental evidence suggests that it serves well as an ideal representation of what happens in real quantum systems.

Table 3.2 summarizes the essential concepts of quantum probability theory. We remark that the transition from classical to quantum probability theory resembles the transition from linear to semidefinite programming.

Concept	Representation	Formula
Probability density	Normalized psd matrix $\boldsymbol{\rho} \in \mathbb{H}_d$	$\boldsymbol{\rho} \succcurlyeq \mathbf{0}$ and $\langle \mathbf{I}, \boldsymbol{\rho} \rangle = 1$
Measurement	Resolution $\{\boldsymbol{H}_{\lambda_i}\}$ of the identity $\mathbf{I}$	$\boldsymbol{H}_{\lambda_i} \succcurlyeq \mathbf{0}$ and $\sum_{i=1}^{m} \boldsymbol{H}_{\lambda_i} = \mathbf{I}$
Born's rule	Trace inner product	$\mathbb{P}\{\lambda_i \mid \boldsymbol{\rho}\} = \langle \boldsymbol{H}_{\lambda_i}, \boldsymbol{\rho} \rangle$

Table 3.2: **Axioms for quantum mechanics.** The structure of quantum mechanics is captured by the real-linear space $\mathbb{H}_d$ endowed with the psd order $\succcurlyeq$ and the identity matrix $\mathbf{I}$.

3.1.3 Aside: Geometric intuition and the Bloch ball

Since the set $\mathcal{S}(\mathbb{H}_d)$ of states is a convex body, we can distinguish points that capture information about its geometry.

- **Extreme points:** A density matrix $\boldsymbol{\rho}$ is an extreme point of $\mathcal{S}(\mathbb{H}_d)$ if and only if $\boldsymbol{\rho}$ has rank one. Equivalently, $\boldsymbol{\rho} = \boldsymbol{u}\boldsymbol{u}^*$ where $\boldsymbol{u} \in \mathbb{C}^d$ is a unit vector. Extreme points are called *pure (quantum) states*, and they generate $\mathcal{S}(\mathbb{H}_d)$ via convex mixtures:

$$\mathcal{S}(\mathbb{H}_d) = \operatorname{conv}\{\boldsymbol{u}\boldsymbol{u}^* : \boldsymbol{u} \in \mathbb{C}^d \text{ and } \|\boldsymbol{u}\| = 1\}.$$

  A pure state is the quantum analog of a classical nonrandom distribution.

- **Barycenter:** The barycenter of $\mathcal{S}(\mathbb{H}_d)$ is the state $\boldsymbol{\rho}_0 = d^{-1}\mathbf{I}$. It is called the *maximally mixed (quantum) state*. The maximally mixed state is the quantum analog of the classical uniform distribution.

For two-dimensional quantum states (called *qubits*), we can construct a beautiful geometric representation, called the *Bloch ball*. This representation helps us visualize the structure of the set of qubits, including the relationships between pure states and the maximally mixed state.

To construct the Bloch ball, we first define the *Pauli matrices*:

$$\boldsymbol{\sigma}_0 = \mathbf{I}; \qquad \boldsymbol{\sigma}_1 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}; \qquad \boldsymbol{\sigma}_2 = \begin{bmatrix} 0 & -\mathrm{i} \\ \mathrm{i} & 0 \end{bmatrix}; \qquad \boldsymbol{\sigma}_3 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.$$

It is straightforward to check that these matrices form a basis of $\mathbb{H}_2$. Consider the parameterized family

$$\boldsymbol{M}(\boldsymbol{r}) = \sum\nolimits_{i=0}^{4} r_i \, \boldsymbol{\sigma}_i \quad \text{where } \boldsymbol{r} = (r_0, r_1, r_2, r_3)^* \in \mathbb{R}^4.$$

We can easily characterize when $\boldsymbol{M}(\boldsymbol{r})$ is a density matrix:

- $\boldsymbol{M}(\boldsymbol{r})$ has unit trace if and only if $r_0 = \frac{1}{2}$.
- $\boldsymbol{M}(\boldsymbol{r})$ is psd if and only if $r_1^2 + r_2^2 + r^3 \leq r_0^2$.

In other terms,

$$\mathcal{S}(\mathbb{H}_2) = \left\{ \tfrac{1}{2}\boldsymbol{\sigma}_0 + \tfrac{1}{2} \sum\nolimits_{i=1}^{3} r_i \, \boldsymbol{\sigma}_i : r_1^2 + r_2^2 + r_3^2 \leq 1 \right\}. \tag{3.2}$$

The set of qubits is parameterized by linear combinations of Pauli matrices whose expansion coefficients $\boldsymbol{r}' = (r_1, r_2, r_3) \in \mathbb{R}^3$ are confined to the unit ball.

The formula (3.2) establishes a one-to-one correspondence between the density matrices of two-dimensional quantum systems and the Euclidean unit ball $\mathbb{S}^2 \subset \mathbb{R}^3$. This is called the *Bloch ball representation*, and it accurately reflects the geometry of $\mathcal{S}(\mathbb{H}_2)$. Indeed,

- The maximally mixed state $\boldsymbol{\rho}_0 = \frac{1}{2}\mathbf{I}$ is associated with the point $\boldsymbol{r}' = \mathbf{0} \in \mathbb{R}^3$, the center of the Bloch ball.
- A density matrix $\boldsymbol{\rho}$ is a pure state if and only if the associated vector $\boldsymbol{r}' \in \mathbb{R}^3$ of expansion coefficients has unit norm. This observation establishes a one-to-one correspondence between pure states (the extreme points of $\mathbb{H}_2$) and unit vectors (the extreme points of $\mathbb{S}^2$).

Figure 3.1 contains an illustration of this correspondence.

Example 3.2 (Stern–Gerlach experiment). Depending on the measurement, a single density matrix can produce both a completely deterministic and a uniformly random outcome distribution. This observation is at the heart of the famous Stern–Gerlach experiment (1922), one of the first demonstrations of genuine quantum behavior.

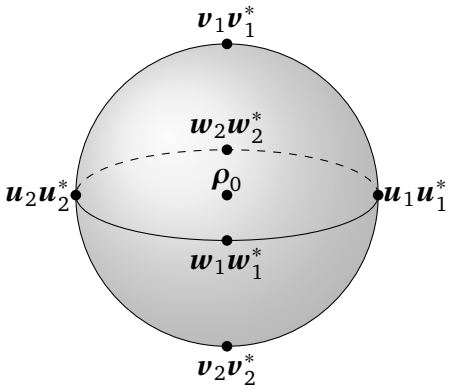


Figure 3.1: **Bloch ball representation of $\mathcal{S}(\mathbb{H}^2)$.** The maximally mixed state $\rho_0 = \frac{1}{2}\mathbf{I}$ lies at the center of the Bloch ball. The surface of the ball is in one-to-one relation with the set of all pure quantum states. Also displayed: Three pairs of mutually orthogonal pure states that are evenly distributed across the boundary of $\mathcal{S}(\mathbb{H}^2)$.

Fix $d = 2$. Define four unit vectors:

$$\begin{aligned} u_1 &= (1,0)^* \quad \text{and} \quad v_1 = \tfrac{1}{\sqrt{2}}(1,1)^*; \\ u_2 &= (0,1)^* \quad \text{and} \quad v_2 = \tfrac{1}{\sqrt{2}}(1,-1)^*. \end{aligned}$$

Then $\{u_1u_1^*, u_2u_2^*\}$ and $\{v_1v_1^*, v_2v_2^*\}$ describe two different quantum measurements. Applied to the pure two-dimensional state $\rho = u_1u_1^*$, these measurements yield radically different outcome distributions:

$$\begin{aligned} \text{Measurement I:} \quad & \mathbb{P}\{1 \,|\, \rho\} = |\langle u_1,\, u_1\rangle|^2 = 1 \quad \text{and} \quad \mathbb{P}\{2 \,|\, \rho\} = 0. \\ \text{Measurement II:} \quad & \mathbb{P}\{2 \,|\, \rho\} = |\langle v_1,\, u_1\rangle|^2 = \tfrac{1}{2} \quad \text{and} \quad \mathbb{P}\{2 \,|\, \rho\} = \tfrac{1}{2}. \end{aligned}$$

We refer to Figure 3.1 for a visualization of the underlying geometry. ∎

3.2 Quantum state tomography

Quantum state tomography is the task of reconstructing the density matrix of a quantum system from measurement data. Quantum tomography is one of the oldest and most fundamental learning problems in quantum information science. Today, quantum tomography is a routine task that is essential for designing, testing, and tuning qubits in our quest to building scalable devices for quantum information processing.

Recall that a density matrix $\rho \in \mathcal{S}(\mathbb{H}_d)$ contains a complete description of a d-dimensional quantum system. Knowledge of the density matrix, therefore, allows us to make predictions about future quantum measurements of an equivalent system. It also contains information about quantum-mechanical aspects of the system. For example, we can compute the *purity* of the system (i.e., the approximate rank of ρ) and the *entanglement* among subsystems of a multipart system (i.e., how strongly the subsystems are correlated).

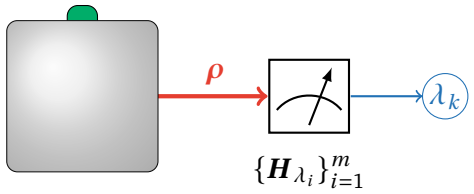


Figure 3.2: **Schematic of quantum state tomography.** A black box (*machine*) that is capable of producing a quantum system with density matrix $\boldsymbol{\rho}$ upon request (*red arrow*). A subsequent quantum measurement (*gauge*) yields a single outcome λ_k (*blue arrow*), but destroys the quantum system. The procedure must be repeated on fresh copies of the state in order to obtain additional information. Quantum state tomography is the task of reconstructing $\boldsymbol{\rho}$ from multiple observed outcomes.

In most settings, the density matrix $\boldsymbol{\rho}$ is not directly accessible. Instead, we obtain indirect information by performing a quantum measurement $\{\boldsymbol{H}_{\lambda_i} : 1 \leq i \leq m\}$. Born's rule (3.1) asserts that data about $\boldsymbol{\rho}$ is encoded in the probability distribution of outcomes (rather than the specific outcome λ_k). Unfortunately, after a measurement is performed, the quantum system ceases to exist. To counter this challenge, we can prepare many copies of the same state, measure each one independently, and combine the information to estimate the distribution of outcomes accurately. Figure 3.2 contains a schematic.

Mathematically, this estimation problem combines interesting aspects of several scientific disciplines, most notably geometry and statistics.

3.2.1 Geometric aspects and measurement design

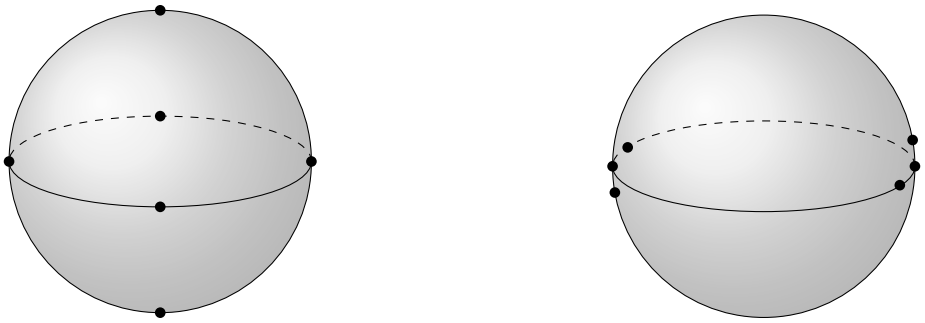
To build up some intuition, we first ignore the statistical aspects of quantum state tomography. Let $\{\boldsymbol{H}_{\lambda_i}\}_{i=1}^m \subset \mathbb{H}_d$ be a fixed measurement. Suppose that we have the capacity to repeatedly perform this measurement on n realizations of an unknown quantum state $\boldsymbol{\rho} \in \mathcal{S}(\mathbb{H}_d)$, where $n \to \infty$. In principle, this operation would allow us to determine the exact (classical) distribution of outcomes:

$$\boldsymbol{p} \in \Delta_m \quad \text{where} \quad p_k = \mathbb{P}\{\lambda_k \,|\, \boldsymbol{\rho}\} = \langle \boldsymbol{H}_{\lambda_k}, \boldsymbol{\rho} \rangle \quad \text{for } 1 \leq k \leq m. \tag{3.3}$$

Thus, Born's rule (3.1) describes a linear map (3.3) between the set $\mathcal{S}(\mathbb{H}_d)$ of density matrices and the set $\Delta_m \subset \mathbb{R}^m$ of classical probability distributions.

In this mathematical idealization, quantum state tomography becomes a linear inverse problem: Recover $\boldsymbol{\rho} \in \mathcal{S}(\mathbb{H}_d)$ from its linear image $\boldsymbol{p} \in \Delta_m$. This task is possible if and only if the linear measurement map (3.3) is injective. The following definition captures this idea.

Definition 3.4 (Tomographic completeness). A quantum measurement system $\{\boldsymbol{H}_{\lambda_i} : 1 \leq i \leq m\} \subset \mathbb{H}_d$ is *tomographically complete* if and only if, for each pair $\boldsymbol{\rho}, \boldsymbol{\sigma} \in \mathcal{S}(\mathbb{H}_d)$ of *distinct* states, there exists an index $k \in \{1, \dots, m\}$ such that $\langle \boldsymbol{H}_{\lambda_k}, \boldsymbol{\rho} \rangle \neq \langle \boldsymbol{H}_{\lambda_k}, \boldsymbol{\sigma} \rangle$.



well conditioned not well-conditioned

Figure 3.3: **Bloch sphere representation of two tomographically complete measurements in $\mathbb{H}_2$.** [*left*] The elements (*points*) of the measurement system are spread out evenly, which results in a well-conditioned measurement map. Indeed, it is a pair of mutually unbiased bases. [*right*] The elements (*points*) of the measurement system cluster at opposite extremes. This measurement is ill-equipped to accurately resolve points on the Bloch sphere in the vicinity of the north and south poles. As a consequence, the associated measurement is not well-conditioned.

A measurement cannot be tomographically complete unless it contains a sufficiently large number of outcomes. Indeed, the number m of potential measurement outcomes must obey

$$m \geq \dim \mathcal{S}(\mathbb{H}_d) + 1 = \dim \mathbb{H}_d = d^2.$$

This is just a basic fact about linear algebra.

Tomographic completeness is not the only property that we require of a measurement system. Indeed, injectivity only implies that the condition number[1] of the measurement map (3.3) is finite. If the condition number κ is large, we will suffer large errors when we try to solve the inverse problem (from a finite amount of data). In contrast, when the condition number $\kappa \approx 1$, the inverse problem can be solved in a stable fashion. Refer to Figure 3.3 for an illustration of two extreme cases, via the Bloch ball representation.

A linear map has the minimal condition number $\kappa = 1$ if and only if the linear map is an isometry. Unfortunately, a quantum measurement map (3.3) can never be tomographically complete and isometric at the same time! The following definition describes the best-conditioned measurement maps that do exist.

Definition 3.5 (Near-isotropic quantum measurement). A quantum measurement system $\{\boldsymbol{H}_{\lambda_i} : 1 \leq i \leq m\} \subset \mathbb{H}_d$ is *near isotropic* when

1. Each element $\boldsymbol{H}_{\lambda_i} = (d/m)\, \boldsymbol{v}_i \boldsymbol{v}_i^*$ where $\boldsymbol{v}_i \in \mathbb{C}^d$ is a unit vector;
2. The measurement has the reconstruction property

$$\frac{1}{m} \sum\nolimits_{i=1}^{m} \langle \boldsymbol{v}_i \boldsymbol{v}_i^*,\ \boldsymbol{X} \rangle\, \boldsymbol{v}_i \boldsymbol{v}_i^* = \frac{1}{(d+1)d} \left(\boldsymbol{X} + (\operatorname{tr} \boldsymbol{X}) \cdot \mathbf{I} \right) \quad \text{for all } \boldsymbol{X} \in \mathbb{H}_d. \tag{3.4}$$

[1] The condition number of a linear map is the ratio between largest and smallest singular value.

The linear map (3.3) associated with a near-isotropic quantum measurement has condition number κ that is bounded independent of the state dimension d. Moreover, the condition number $\kappa \to 1$ as the state dimension $d \to \infty$.

In other research areas, you may encounter a system $\{\boldsymbol{v}_1, \dots, \boldsymbol{v}_m\} \subset \mathbb{C}^d$ of unit vectors that satisfies (3.4). In approximation theory, these systems are interpreted as quadrature rules for polynomials on the complex unit sphere, and they are known as (complex projective) 2-designs. In frame theory, these systems are called tight fusion frames.

There are many interesting constructions of near-isotropic quantum measurements that arise from these connections.

Example 3.3 (Near-isotropic quantum measurements). The following quantum measurement systems are near-isotropic.

1. The *uniform measurement* is the infinite family $\{d\,\boldsymbol{v}\boldsymbol{v}^*\}$ of all rescaled rank-one projectors $d\,\boldsymbol{v}\boldsymbol{v}^*$, endowed with the unique rotation-invariant probability measure $\mathrm{d}\boldsymbol{v}$ on the complex unit sphere.
2. The union of $d + 1$ mutually unbiased bases[2] forms a set of $m = (d + 1)d$ unit vectors that obey (3.4). Explicit constructions of these families are known when the dimension d is a prime power.
3. A set of $m = d^2$ equiangular lines in $\mathbb{C}^d$ also obeys (3.4). Zauner's conjecture states that such configurations should exist in any dimension d. A complete proof of this conjecture is an important open problem.

In theory, each of these measurement systems provides an effective way of acquiring information about a quantum state. But these measurements are not always realizable in practical settings. Constructing well-conditioned, implementable measurements is a major challenge in quantum engineering. ∎

3.2.2 Statistical aspects and convergence

So far, we have considered a mathematically ideal version of quantum state tomography in which we measure an infinite number of realizations of the same state. In practice, the number n of samples is necessarily finite, so we cannot determine the probability distribution over the measurement outcomes exactly. As a consequence, we cannot expect to recover an unknown density matrix exactly.

Nevertheless, we can obtain an accurate approximation of the state with high probability, provided that we can perform a sufficient number of measurements. To assess the accuracy of an estimate $\hat{\boldsymbol{\rho}} \in \mathbb{H}_d$ of a state $\boldsymbol{\rho} \in \mathcal{S}(\mathbb{H}_d)$, we typically use the Schatten 1-norm $\|\hat{\boldsymbol{\rho}} - \boldsymbol{\rho}\|_1$. The Schatten 1-norm, also known as the *trace norm*, is the quantum analog of the total-variation distance that arises in classical probability. Furthermore, this error measure has a natural operational interpretation in terms of quantum hypothesis testing.

Definition 3.6 (Sample complexity). Fix parameters $\varepsilon, \delta \in (0, 1)$ and a rank $1 \le r \le d$. Let $\boldsymbol{\rho} \in \mathbb{H}_d$ be an unknown state with rank r. Perform the same quantum measurement

[2] Two orthonormal bases $\{\boldsymbol{b}_1, \dots, \boldsymbol{b}_d\}$ and $\{\boldsymbol{c}_1, \dots, \boldsymbol{c}_d\} \subset \mathbb{C}^d$ are mutually unbiased if $|\langle \boldsymbol{b}_i, \boldsymbol{c}_j \rangle|^2 = d^{-1}$ for all $1 \le i, j \le d$. The standard basis and the discrete Fourier basis provide an instructive example.

on n realizations of the state, and construct a tomographic estimate $\hat{\boldsymbol{\rho}}_n \in \mathbb{H}_d$. The *sample complexity* of this family of estimators is the minimum number n required to estimate the state with high probability:

$$\|\hat{\boldsymbol{\rho}}_n - \boldsymbol{\rho}\|_1 \leq \varepsilon \quad \text{with probability } 1 - \delta.$$

The sample complexity will typically depend on the rank r.

Methods from quantum information theory lead to a rigorous lower bound on the sample complexity of *any* tomographic estimation procedure [Haa+17].

Theorem 3.7 (Haah et al. 2017 — informal). *Any* tomographic estimator based on repeating the same measurement has sample complexity

$$n \gtrsim r^2 d \varepsilon^{-2} \log(1/\delta).$$

3.3 Quantum state tomography via matrix sampling

We are going to present and analyze a simple, yet powerful, estimation technique for quantum state tomography based on matrix sampling. To motivate the approach, we first consider a problem in classical probability.

3.3.1 Estimating the bias of a coin

A classical analog of quantum state estimation is the problem of estimating the bias in a coin by flipping it repeatedly.

A coin is a two-dimensional classical random variable that is described by a single parameter, the *bias* $p \in [0, 1]$. The two outcomes follow the distribution $\mathbb{P}\{\text{heads}\} = p$ and $\mathbb{P}\{\text{tails}\} = 1 - p$. How can we estimate the bias p by repeatedly tossing the coin? The simplest approximation procedure is based on a simple and intuitive decision rule. Toss the coin once and set

$$\hat{p} = \begin{cases} 1, & \text{if heads;} \\ 0, & \text{if tails.} \end{cases}.$$

In general, this is a terrible estimator. But it does have the virtue of being unbiased:

$$\mathbb{E}\,\hat{p} = p \times 1 + (1 - p) \times 0 = p.$$

Instead, we toss the coin n times, form the estimators $\hat{p}_1, \ldots, \hat{p}_n$, and construct the empirical average:

$$\bar{p}_n = n^{-1} \sum_{i=1}^{n} \hat{p}_i.$$

The empirical average will converge to the true bias of the coin.

To verify this claim and obtain a convergence rate, just apply Theorem 2.1 with $d = 1$. For each $t \in [0, 1]$,

$$\mathbb{P}\{|\bar{p}_n - p| \geq t\} \leq 2 \exp\left(\frac{-nt^2/2}{p + 2t/3}\right) \leq 2 \exp\left(-\tfrac{3}{10} nt^2\right).$$

Therefore, for any parameters $\varepsilon, \delta \in (0, 1)$, if we perform

$$n \geq \tfrac{10}{3}\varepsilon^{-2} \log(1/(2\delta))$$

independent coin tosses, then the sample average satisfies the error bound $|\bar{p}_n - p| < \varepsilon$ with probability at least $1 - \delta$.

3.3.2 The matrix sampling estimator

The simple coin tossing example can readily be generalized to quantum state tomography in $d \geq 2$ dimensions. Construct a quantum system with unknown density matrix $\boldsymbol{\rho} \in \mathcal{S}(\mathbb{H}_d)$, and perform the near-isotropic quantum measurement

$$\{\boldsymbol{H}_{\lambda_i} = (d/m)\, \boldsymbol{v}_i \boldsymbol{v}_i^* : 1 \leq i \leq m\}.$$

When reading the measurement outcome, set

$$\boldsymbol{R} = \begin{cases} (d+1)\, \boldsymbol{v}_1 \boldsymbol{v}_1^* - \mathbf{I}, & \text{if we observe outcome } \lambda_1; \\ \qquad\vdots & \\ (d+1)\, \boldsymbol{v}_m \boldsymbol{v}_m^* - \mathbf{I}, & \text{if we observe outcome } \lambda_m. \end{cases} \tag{3.5}$$

Born's rule (3.1) and the geometric properties of near-isotropic measurements (3.4) ensure that the quantum estimator (3.5) is correct in expectation:

$$\begin{aligned} \mathbb{E}(\boldsymbol{R} + \mathbf{I}) &= \sum\nolimits_{i=1}^{m} \mathbb{P}\{\lambda_i \,|\, \boldsymbol{\rho}\} \cdot (d+1)\, \boldsymbol{v}_i \boldsymbol{v}_i^* \\ &= \frac{(d+1)d}{m} \sum\nolimits_{i=1}^{m} \operatorname{tr}(\boldsymbol{v}_i \boldsymbol{v}_i^* \boldsymbol{\rho})\, \boldsymbol{v}_i \boldsymbol{v}_i^* = \boldsymbol{\rho} + \mathbf{I}. \end{aligned} \tag{3.6}$$

We also remark that $\boldsymbol{R}$ has trace one, but it need not be psd.

We repeat this estimation procedure n times, for n copies of the quantum system, and we construct the sample average:

$$\bar{\boldsymbol{R}}_n = \frac{1}{n} \sum\nolimits_{i=1}^{n} \boldsymbol{R}_i \quad \text{where } \boldsymbol{R}_i \text{ are iid copies of } \boldsymbol{R}.$$

The sample average has trace one, and it is an unbiased estimator of the state. On the other hand, it is not always psd. See Figure 3.4 for an illustration of the convergence of this sequence of estimates.

This construction is formally similar to the one in Lecture 2, but let us point out a major conceptual difference. Before, we designed an algorithm that makes random choices to construct random matrices that approximate a kernel matrix. In quantum state tomography, the estimator (3.5) produces independent random matrices because of the laws of quantum mechanics.

3.3.3 Sample complexity of the sample average

We quickly derive the convergence rate of the sample average using Theorem 2.1.

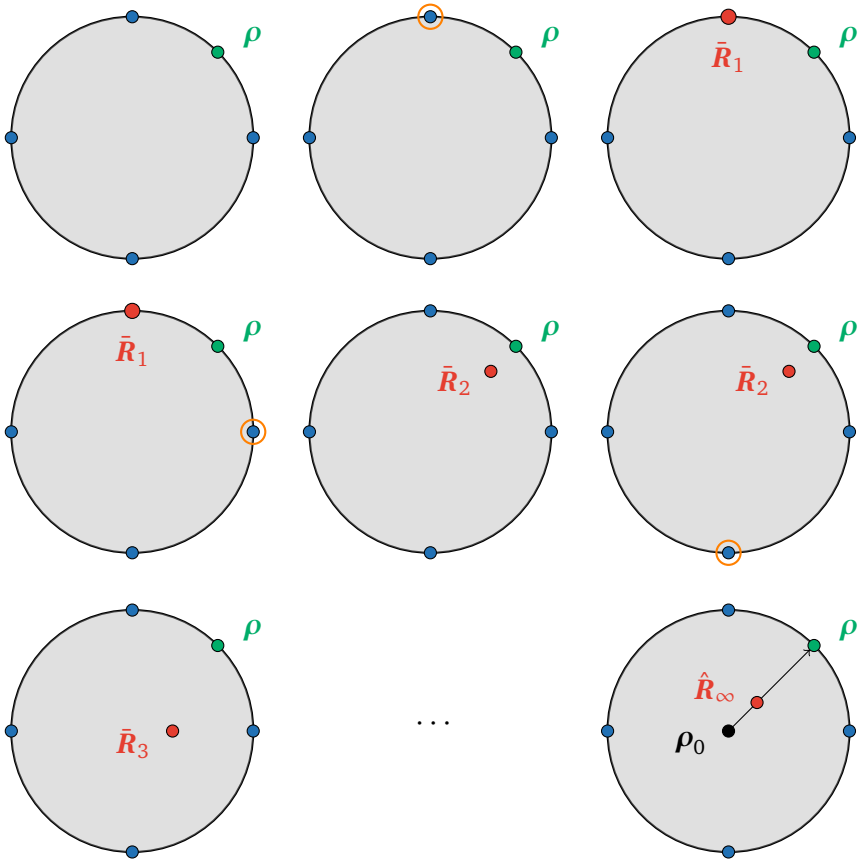


Figure 3.4: **Convergence of a naïve tomography estimator.** Perform a near-isotropic measurement (*blue points*) on an unknown density matrix $\boldsymbol{\rho}$ (*green*). Upon receiving a certain outcome (*orange circle*), we estimate $\boldsymbol{\rho}$ by the measurement element associated with this outcome (*red*). Repeat this procedure n times and construct the empirical average $\bar{\boldsymbol{R}}_n$. Convex mixing properties imply that this estimator is pushed inside the set of quantum states (*convex combination*). But it maintains the correct direction in the limit of many repetitions. This illustration is exact if we restrict attention to the equatorial plane of the Bloch ball representation of $\mathcal{S}(\mathbb{H}_2)$.

Proposition 3.8 (Quantum state tomography with sample averages). Suppose that we have access to n (unentangled) realizations of a quantum system with density matrix $\boldsymbol{\rho} \in \mathcal{S}(\mathbb{H}_d)$. Perform a near-isotropic quantum measurement on the ith realization, and construct the estimator $\boldsymbol{R}_i \in \mathbb{H}_d$ based on the decision rule (3.5). Then the *matrix sample average* $\bar{\boldsymbol{R}}_n = n^{-1} \sum_{i=1}^n \boldsymbol{R}_i$ obeys the error estimate

$$\mathbb{P}\left\{\|\bar{\boldsymbol{R}}_n - \boldsymbol{\rho}\| \geq t\right\} \leq 2d \exp\left(\frac{-3nt^2}{16d}\right).$$

This formula is valid for all $t \in [0, 1]$.

Proof. Let $\boldsymbol{R}$ be the estimator (3.5). First, compute the upper bound

$$B = \sup \|\boldsymbol{R}\| = \max_{i=1,\dots,m} \|(d+1)\,\boldsymbol{v}_i \boldsymbol{v}_i^* - \mathbf{I}\| = d.$$

Another short calculation shows that the per-sample second moment satisfies

$$m_2(\boldsymbol{R}) = \|\mathbb{E}\,\boldsymbol{R}^2\| = \|(d-1)\,\boldsymbol{\rho} + d\,\mathbf{I}\| \leq 2d.$$

We leave the details as an easy exercise.

Now, quantum measurements of unentangled quantum systems ensures are statistically independent. Therefore, the random matrices $\boldsymbol{R}_i$ are independent copies of the random matrix $\boldsymbol{R}$. Theorem 2.1 implies that the matrix sample estimator $\bar{\boldsymbol{R}}_n = n^{-1} \sum_{i=1}^n \boldsymbol{R}_i$ concentrates sharply around its expectation $\boldsymbol{\rho}$. For $t \geq 0$,

$$\mathbb{P}\left\{\|\bar{\boldsymbol{R}}_n - \boldsymbol{\rho}\| \geq t\right\} \leq 2d \exp\left(\frac{-nt^2/2}{m_2(\boldsymbol{R}) + 2Bt/3}\right) \leq 2d \exp\left(\frac{-3nt^2}{16d}\right).$$

This is what we needed to show. ∎

3.3.4 Projection onto the set of quantum states

Proposition 3.8 equips the matrix sample average estimator $\bar{\boldsymbol{R}}_n$ with a rigorous guarantee that it converges to the unknown density matrix $\boldsymbol{\rho} \in \mathcal{S}(\mathbb{H}_d)$. For $\tau \in (0, 1)$, a total of $n \gtrsim d \log(d)/\tau^2$ measurement repetitions are sufficient to ensure that $\|\bar{\boldsymbol{R}}_n - \boldsymbol{\rho}\| \leq \tau$ with high probability. Although powerful, this statement has two drawbacks:

1. The matrix sample estimator $\bar{\boldsymbol{R}}_n$ is typically not psd. We therefore estimate the state $\boldsymbol{\rho}$ by something that is not itself a state.
2. Accuracy is reported in operator norm distance, rather than trace-norm distance.

Surprisingly, both drawbacks can be overcome by a single refinement. Just replace the sample matrix estimator by the closest density matrix, computed with respect to the Frobenius norm:

$$\hat{\boldsymbol{\rho}}_n = \operatorname*{arg\,min}_{\boldsymbol{\sigma} \in \mathcal{S}(\mathbb{H}_d)} \|\boldsymbol{\sigma} - \bar{\boldsymbol{R}}_n\|_{\mathrm{F}}. \tag{3.7}$$

We call this estimator the *projected matrix sample average*.

Intuitively, the projection onto the density matrices should decrease the distance between the estimator and target state. The following technical result makes this claim precise.

Lemma 3.9 Fix a rank-r density matrix $\boldsymbol{\rho} \in \mathcal{S}(\mathbb{H}_d)$ and a matrix $\boldsymbol{M} \in \mathbb{H}_d$ with trace one. Then the closest density matrix $\boldsymbol{\sigma}$ to $\boldsymbol{M}$ necessarily obeys

$$\|\boldsymbol{\rho} - \boldsymbol{\sigma}\|_1 \leq 4r \, \|\boldsymbol{\rho} - \boldsymbol{M}\|.$$

Proof sketch. The difference $\boldsymbol{X} = \boldsymbol{\rho} - \boldsymbol{\sigma}$ is a traceless self-adjoint matrix. Moreover, the positive part of $\boldsymbol{X}$ has rank no greater than r because both $\boldsymbol{\rho}$ and $\boldsymbol{\sigma}$ are psd and $\operatorname{rank}(\boldsymbol{\rho}) = r$. Let $\boldsymbol{P}_{\pm} \in \mathbb{H}_d$ denote the orthogonal projectors onto the positive and negative parts of $\boldsymbol{X}$. Then

$$\|\boldsymbol{X}\|_1 = \langle \boldsymbol{P}_+, \, \boldsymbol{X} \rangle - \langle \boldsymbol{P}_-, \, \boldsymbol{X} \rangle = 2 \langle \boldsymbol{P}_+, \, \boldsymbol{X} \rangle,$$

where the last equation follows from the fact that $\boldsymbol{X}$ is traceless. The matrix Hölder inequality asserts that $\langle \boldsymbol{P}_+, \, \boldsymbol{X} \rangle \leq \|\boldsymbol{P}_+\|_1 \, \|\boldsymbol{X}\|$. Therefore,

$$\|\boldsymbol{\rho} - \boldsymbol{\sigma}\|_1 = \|\boldsymbol{X}\|_1 \leq 2 \operatorname{tr}(\boldsymbol{P}_+) \, \|\boldsymbol{\rho} - \boldsymbol{\sigma}\| \leq 2r \, \|\boldsymbol{\rho} - \boldsymbol{\sigma}\|.$$

Indeed, the range of the orthogonal projector $\boldsymbol{P}_+$ has dimension at most r. The result follows once we establish that $\|\boldsymbol{\rho} - \boldsymbol{\sigma}\| \leq 2 \, \|\boldsymbol{\rho} - \boldsymbol{M}\|$. This relation follows from the assumption that $\boldsymbol{M}$ has unit trace, but the proof is somewhat less transparent. ∎

The following convergence bound is an immediate consequence of Proposition 3.8 and Lemma 3.9.

Theorem 3.10 (Projected sample average estimator). Suppose that we perform near-isotropic quantum measurements on identical copies of a quantum system that has the rank-r density matrix $\boldsymbol{\rho}$. Then the projected matrix sample average (3.7) obeys

$$\mathbb{P}\left\{ \|\hat{\boldsymbol{\rho}}_n - \boldsymbol{\rho}\|_1 \geq t \right\} \leq 2d \exp\left(\frac{-3nt^2}{256 r^2 d} \right)$$

The probability bound is valid for all $t \geq 0$.

In short, the matrix Bernstein inequality leads quickly to a strong error bound on the projected sample average estimator of a quantum state. The following observation is an immediate consequence of Theorem 3.10.

Corollary 3.11 Fix a rank-r density matrix $\boldsymbol{\rho} \in \mathcal{S}(\mathbb{H}_d)$. Choose parameters $\varepsilon, \delta \in (0, 1)$. Then a total of

$$n \geq 86 r^2 d \varepsilon^{-2} \left(\log(2d) + \log(1/\delta) \right)$$

measurement repetitions (samples) are sufficient to guarantee that the projected sample average estimator obeys $\|\hat{\boldsymbol{\rho}}_n - \boldsymbol{\rho}\|_1 \leq \varepsilon$ with probability at least $1 - \delta$.

We conclude that the projected sample average estimator almost saturates the fundamental lower bound (Theorem 3.7) on the sample complexity of *any* quantum state tomography procedure. Moreover, the performance is optimal up to a constant factor in the regime where the probability of success is at least $1 - d^{-1}$!

3.3.5 Generalization: Projected least squares

The matrix sample average estimator for near-isotropic quantum measurements is a special case of a general and practical procedure for quantum state tomography, called *projected least squares*. Here is a summary of this approach:

1. Fix a tomographically complete measurement $\{\boldsymbol{H}_{\lambda_i} : 1 \le i \le m\}$.
2. Estimate the probabilities $p_i = \mathbb{P}\{\lambda_i \,|\, \boldsymbol{\rho}\}$ by *frequencies*. That is, prepare n identical realizations of the quantum system, measure them separately, and set

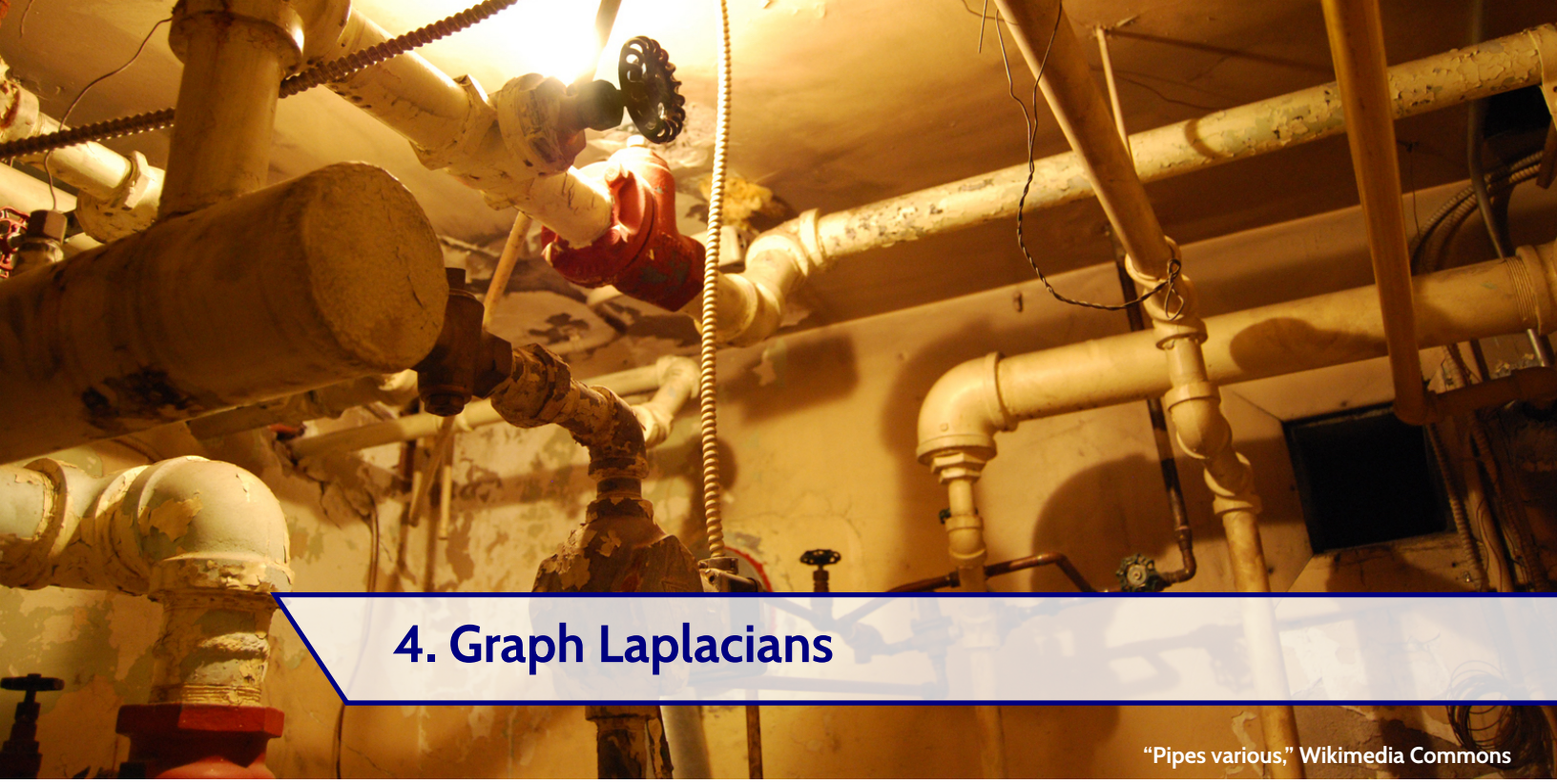
$$f_i^{(n)} = \frac{\#\text{ number of times outcome } \lambda_i \text{ was observed}}{\text{total number of measurements } n}.$$

3. Construct the least-squares estimator that results from replacing the true probabilities in Born's rule (3.1) by the frequency approximations:

$$\bar{\boldsymbol{R}}_n = \operatorname*{arg\,min}_{\boldsymbol{X} \in \mathbb{H}_d} \quad \sum\nolimits_{i=1}^{m} |f_i^{(n)} - \langle \boldsymbol{H}_{\lambda_i}, \boldsymbol{X} \rangle|^2.$$

4. Compute the Frobenius-norm projection of $\bar{\boldsymbol{R}}_n$ onto the set $\mathcal{S}(\mathbb{H}_d)$ of quantum states.

This procedure also results in a near-optimal quantum state estimator. As above, the analysis relies on the matrix Bernstein inequality. The main difference is that the solution to the linear inverse problem has a more complicated form when the measurement is not near-isotropic.



4. Graph Laplacians

"Pipes various," Wikimedia Commons

> This lecture contains the fundamentals of spectral graph theory and harmonic analysis on graphs. The presentation is inspired by Dan Spielman's Fall 2018 course on spectral graph theory [Spi], Yuval Wigderson's notes on harmonic functions on graphs [Wig], and Rasmus Kyng's dissertation [Kyn17]. Any errors are my own.

A combinatorial graph encodes pairwise relationships among a family of objects. Graphs have intrinsic mathematical interest, as well as numerous computational applications. This lecture introduces the concept of a multigraph and the associated Laplacian matrix. The Laplacian encodes structural properties of the multigraph, and it can be understood with physical analogies to electrical networks.

Laplacian matrices play a role in learning methods based on harmonic analysis on manifolds. They also arise from the discretization of elliptic PDEs. The ultimate goal of this course is to present an efficient algorithm for solving a linear system in a graph Laplacian matrix, which can be used for both of the applications mentioned in this paragraph.

4.1 Multigraph basics

We will be working with (undirected) multigraphs, which are a lot like graphs, except that there may be many edges connecting a pair of vertices. This level of generality is important for us, so we must suffer the extra complexity.

4.1.1 Undirected multigraphs

Let V be a set of n points, called *vertices*. The letters u and v will denote vertices. We may as well assume that $\mathsf{V} = \{1, \dots, n\}$, which allows us place the vertices in order.

A *multiedge* is an unordered pair $e = \{u, v\}$ of two distinct vertices $u, v \in \mathsf{V}$. A multiedge represents an undirected link between the two vertices, and we forbid loops that connect a vertex to itself. It is convenient to abbreviate $e = uv = vu$ for any multiedge connecting u and v. The notations $u \in e$ and $e \ni u$ both mean that the multiedge e contains the vertex u. We also say that e is *incident* on u.

We assign each multiedge a unique label so we can tell it apart from other multiedges between the same two vertices. At the risk of some confusion, we completely suppress this label from the notation.

An (undirected) *multigraph* G consists of a ground set V of vertices, along with a family E of multiedges. The letter $m = |\mathsf{E}|$ will refer to the total number of multiedges.

Somewhat abusively, we may write either $e \in \mathsf{E}$ or $e \in \mathsf{G}$ to indicate that the multigraph contains the multiedge e. (There are further notational abuses to come!)

We also equip with the multigraph G with a nonnegative weight function $w : \mathsf{E} \to \mathbb{R}_{++}$ that assigns a strictly positive value to each multiedge. Note that each multiedge joining a single pair of vertices can have a distinct weight.

We will always be working with the same ground set V of vertices, but there will be many multigraphs floating around. Therefore, is it often useful to qualify our notation by specifying a multigraph. For example, we may write $w_{\mathsf{G}}(e)$ or $w(e, \mathsf{G})$ to refer to the weight of a multiedge in the multigraph G.

4.1.2 Connected components

A vertex u is a *neighbor* of a vertex v if the multigraph contains at least one multiedge $e = uv$ linking the vertices u and v. We write $u \sim v$ or $v \sim u$ to indicate that u and v are neighbors.

We can iterate the neighbor relation to obtain *multi-hop neighborhoods* of a vertex. For a vertex $u \in \mathsf{V}$, iteratively define

$$\mathsf{N}^0(u) = \{u\} \quad \text{and} \quad \mathsf{N}^k(u) = \{v' \in \mathsf{V} : v' \sim v \text{ and } v \in \mathsf{N}^{k-1}(u)\} \quad \text{for } k \in \mathbb{N}.$$

The set $\mathsf{N}^k(u)$ contains the vertices that are reachable from u by traversing exactly k multiedges. It is common to abbreviate $\mathsf{N}(u) = \mathsf{N}^1(u)$.

The *connected component* $\mathsf{N}^\infty(u)$ of a vertex u is the set of all vertices that are reachable from u via the neighbor relation:

$$\mathsf{N}^\infty(u) = \bigcup\nolimits_{k=0}^{\infty} \mathsf{N}^k(u).$$

Every multigraph can be partitioned into a disjoint family of connected components. The relation $\mathsf{N}^\infty(u) = \mathsf{V}$ means that every vertex in the multigraph is reachable from u. In the latter case, every vertex is reachable from every other vertex, and we say that the multigraph is *connected*.

> From now on, we will assume that the multigraph G is connected.

4.1.3 Multidegree and total weight

The *degree*, $\deg(u)$, of a vertex u in the multigraph G is the total number of multiedges incident on u. That is,

$$\deg(u) = \deg(u, \mathsf{G}) = |\{e \in \mathsf{G} : e \ni u\}|.$$

Note that the multidegree of u need not coincide with the number of vertices that neighbor u.

The *total weight* $w(u)$ of a vertex u in the multigraph G is the the sum of the weights of the multiedges that are incident on u. That is,

$$w(u) = w_{\mathsf{G}}(u) = \sum\nolimits_{e \in \mathsf{G},\, e \ni u} w_{\mathsf{G}}(e).$$

Take care that the weight function has a different definition when applied to vertices and edges.

4.1.4 Interpretation: Plumbing

We can interpret a multigraph $\mathsf{G} = (\mathsf{V}, \mathsf{E}, w)$ as a plumbing network that connects the fixtures listed in V with the pipes listed in E. There may be many pipes connecting the same two fixtures. The weight $w(e)$ associated with a pipe e increases with the "size" of the pipe.

For later reference, recall that the rate of flow along a pipe is proportional to the "size" of the pipe times the difference in pressure at the two endpoints. (The size of a circular pipe is the fourth power of the radius divided by the length.) This is called the Hagen–Poiseuille law.

4.1.5 Interpretation: Resistor networks

We can interpret a multigraph $\mathsf{G} = (\mathsf{V}, \mathsf{E}, w)$ as a wiring diagram that connects the terminals V with the wires E. There may be many wires connecting the same two terminals in parallel. The weight $w(e)$ of a wire e is proportional to the electrical conductance of the wire. The weight $w(e)$ is inversely proportional to the electrical resistance.

For later reference, we recall Ohm's law: $V = IR$. In words, the difference (V) in voltage at two terminals is proportional to the electrical current (I) flowing between the terminals times the electrical resistance (R) of the wire.

4.1.6 Example: A random walk

There is a natural construction of a random walk on G. Let $u_0 \in \mathsf{V}$ be the initial vertex. At each time $k \in \mathbb{N} \cup \{0\}$, we are at vertex u_k, and we draw the next vertex u_{k+1} in the walk at random according to the probability distribution

$$\mathbb{P}\{u_{k+1} = v \mid u_k = u\} = \frac{1}{w(u)} \sum\nolimits_{e = uv \in \mathsf{G}} w(e) \quad \text{for each } v \in \mathsf{N}(u).$$

Each multiedge of the form $e = uv$ appears once in the sum! In other words, we randomly choose one of the multiedges incident on u with probability in proportion to

its weight, and we traverse this edge to arrive at a new vertex v. This process repeats indefinitely.

The transition matrix $\boldsymbol{Q}$ of the random walk is called the *random walk normalized Laplacian*, and it is obtained by diagonal reweighting of the ordinary Laplacian:

$$\boldsymbol{Q} = \operatorname{diag}(w(u) : u \in \mathsf{V})^{-1} \boldsymbol{L}.$$

One can understand many features of the random walk by studying the eigenvalues and eigenvectors of the random walk normalized Laplacian. But this is a subject for another day.

4.2 Laplacian basics

Every multigraph is associated with a psd matrix, called the *Laplacian*. The properties of this matrix, as a linear operator, are intertwined with the structure of the multigraph.

4.2.1 The Laplacian of a multigraph

Let $e = uv$ be a multiedge connecting distinct vertices $u, v \in \mathsf{V}$. The *elementary Laplacian* induced by the multiedge e is the matrix

$$\boldsymbol{\Delta}_e = \boldsymbol{\Delta}_{uv} = (\boldsymbol{\delta}_u - \boldsymbol{\delta}_v)(\boldsymbol{\delta}_u - \boldsymbol{\delta}_v)^* \in \mathbb{H}_{\mathsf{V}}.$$

Recall that $\boldsymbol{\delta}_u$ denotes the standard basis vector at vertex u. Observe that the elementary Laplacian is a psd matrix. In addition, the null space of the elementary Laplacian contains the constant vector $\mathbf{1} \in \mathbb{R}^{\mathsf{V}}$.

Definition 4.1 (Graph Laplacian). The *Laplacian* of the multigraph G is the matrix

$$\boldsymbol{L} = \boldsymbol{L}_{\mathsf{G}} = \sum\nolimits_{e \in \mathsf{G}} w(e)\, \boldsymbol{\Delta}_e \in \mathbb{H}_{\mathsf{V}}. \tag{4.1}$$

The Laplacian $\boldsymbol{L}$ is a psd matrix because it is a nonnegative sum of psd matrices. For distinct vertices $u, v \in \mathsf{V}$, the uv off-diagonal entry of the Laplacian records (the negative of) the total weight of all the multiedges connecting u and v:

$$(\boldsymbol{L})_{uv} = -\sum\nolimits_{e=uv} w(e).$$

Meanwhile, the diagonal of the Laplacian records the total weight of each vertex of the graph:

$$w(u) = (\boldsymbol{L})_{uu} = \sum\nolimits_{e \ni u} w(e) \quad \text{for each } u \in \mathsf{V}.$$

The diagonal and off-diagonal entries are related as

$$w(u) = (\boldsymbol{L})_{uu} = -\sum\nolimits_{v \neq u} (\boldsymbol{L})_{uv}.$$

The last display is another statement of the fact that $\boldsymbol{L}\mathbf{1} = \mathbf{0}$.

Exercise 4.1 Assume that G is a connected multigraph. Prove that $\operatorname{null}(\boldsymbol{L}_{\mathsf{G}}) = \operatorname{lin}\{\mathbf{1}\}$.

Exercise 4.2 Consider a symmetric matrix $\boldsymbol{M} \in \mathbb{H}_{\mathsf{V}}$ for which

1. $\boldsymbol{M}$ has nonnegative diagonal entries;
2. $\boldsymbol{M}$ has nonpositive off-diagonal entries;
3. $\boldsymbol{M}\mathbf{1} = \mathbf{0}$.

Show that $\boldsymbol{M}$ is the Laplacian of some (multi)graph. In particular, the class of Laplacian matrices forms a convex cone. (That is, the class is closed under addition and nonnegative scaling.)

4.2.2 Correspondence between multigraphs and Laplacians

Each multigraph determines a unique Laplacian matrix, but the converse is not true. For the purposes of our presentation, we will elide this point by treating the Laplacian of the multigraph as a sum over multiedges. Moreover, we usually regard the multigraph and the Laplacian as interchangeable.

Let us take a minute to justify this decision more rigorously. We will construct a pair of matrices that are closely related to the Laplacian and that completely determine the multigraph. This approach is also useful for implementing algorithms.

To that end, let us enumerate the multiedges in the multigraph: $e_1, e_2, \dots, e_m \in \mathsf{E}$. The ordering is arbitrary, but fixed. The signed vertex–multiedge adjacency matrix $\boldsymbol{A} \in \mathbb{R}^{\mathsf{V}\times\mathsf{E}}$ encodes the connectivity of the graph. The jth multiedge $e_j = u_j v_j$ determines the jth column of the matrix:

$$\boldsymbol{a}_{:j} = \boldsymbol{\delta}_{u_j} - \boldsymbol{\delta}_{v_j} \quad \text{where} \quad u_j < v_j \quad \text{for each } e_j \in \mathsf{E}.$$

The ordering is chosen for concreteness, but it is unimportant. Second, introduce a nonnegative diagonal matrix $\boldsymbol{W} \in \mathbb{H}_{\mathsf{E}}$ that encodes the weights in the obvious way:

$$w_{jj} = w(e_j) \quad \text{for each } e_j \in \mathsf{E}.$$

Together, $\boldsymbol{A}$ and $\boldsymbol{W}$ contain all of the data about the graph.

These two matrices provide another construction of the Laplacian of the multigraph:

$$\boldsymbol{L} = \boldsymbol{A}\boldsymbol{W}\boldsymbol{A}^*.$$

This gives another precise sense to our identification of the Laplacian with a sum of multiedges.

4.2.3 Projectors and pseudoinverses

We will be keenly interested in solving linear systems involving the Laplacian matrix $\boldsymbol{L}$ of a multigraph G. This requires some care because the Laplacian is singular.

Definition 4.2 (Range projector and pseudoinverse). Let G be a connected multigraph with Laplacian $\boldsymbol{L}$. The orthogonal projector $\boldsymbol{P} \in \mathbb{H}_{\mathsf{V}}$ onto the range of $\boldsymbol{L}$ is the matrix

$$\boldsymbol{P} = \mathbf{I} - |\mathsf{V}|^{-1}\mathbf{1}\mathbf{1}^*.$$

The *pseudoinverse* $\boldsymbol{L}^\dagger \in \mathbb{H}_{\mathsf{V}}$ is the unique psd matrix that satisfies

$$\boldsymbol{L}\boldsymbol{L}^\dagger = \boldsymbol{P} \quad \text{and} \quad \operatorname{range}(\boldsymbol{L}^\dagger) = \operatorname{range}(\boldsymbol{L}).$$

The next two results are easy consequences of the definitions.

Exercise 4.3 (Laplacian pseudoinverse). For a connected multigraph G, the Laplacian matrix $\boldsymbol{L}$ and the range projector $\boldsymbol{P}$ enjoy the following relationships:

1. $\boldsymbol{L}\boldsymbol{P} = \boldsymbol{P}\boldsymbol{L}$.
2. $\boldsymbol{L}^\dagger \boldsymbol{L} = \boldsymbol{P}$.
3. $\boldsymbol{L}^\dagger \boldsymbol{L} \boldsymbol{L}^\dagger = \boldsymbol{L}^\dagger$.
4. $\boldsymbol{L}\boldsymbol{L}^\dagger \boldsymbol{L} = \boldsymbol{L}$.

Exercise 4.4 (Laplacian linear systems). Let G be a connected multigraph with Laplacian matrix $\boldsymbol{L}$. Suppose that $\boldsymbol{f} \in \mathbb{R}^{\mathsf{V}}$ satisfies the orthogonality relation $\mathbf{1}^* \boldsymbol{f} = 0$. Then

$$\boldsymbol{L}\boldsymbol{x} = \boldsymbol{f} \quad \text{and} \quad \mathbf{1}^* \boldsymbol{x} = 0 \quad \text{if and only if} \quad \boldsymbol{x} = \boldsymbol{L}^\dagger \boldsymbol{f}.$$

4.2.4 The Dirichlet form

The Laplacian induces a quadratic form, called the *Dirichlet form*:

$$\|\boldsymbol{x}\|_{\boldsymbol{L}}^2 = \boldsymbol{x}^* \boldsymbol{L} \boldsymbol{x} = \sum\nolimits_{e=uv} w(e) \, (x_u - x_v)^2 \quad \text{for } \boldsymbol{x} \in \mathbb{R}^{\mathsf{V}}.$$

Note that each multiedge of the form $e = uv$ appears once in the sum! The associated pseudonorm is called the *Dirichlet energy*:

$$\|\boldsymbol{x}\|_{\boldsymbol{L}} = (\boldsymbol{x}^* \boldsymbol{L} \boldsymbol{x})^{1/2} \quad \text{for } \boldsymbol{x} \in \mathbb{R}^{\mathsf{V}}.$$

The Dirichlet norm of a vector $\boldsymbol{x} \in \mathbb{R}^{\mathsf{V}}$ reflects its smoothness with respect to the graph structure.

The Dirichlet energy has various physical interpretations that are useful for constructing graph embeddings. The Dirichlet energy also provides a natural way to quantify the error in solving a linear system in the Laplacian matrix.

4.2.5 Example: Laplacians and cuts

Here is a simple connection between the Dirichlet form and the combinatorial properties of a graph. A *cut* in a multigraph is a subset U of the vertices. The *weight* of a cut is the total weight of the multiedges that cross the cut:

$$\text{weight}(\mathsf{U}) = \sum\nolimits_{e=uv;\, u \in U;\, v \notin U} w(e).$$

Note that each multiedge $e = uv$ in the multigraph appears at most once in the sum, with the orientation $u \in \mathsf{U}$ and $v \notin \mathsf{U}$. The Laplacian allows us to express the weight of a cut. Evaluate the Dirichlet form at the indicator vector of the cut to obtain the weight of the cut:

$$\text{weight}(\mathsf{U}) = \|\boldsymbol{\chi}_{\mathsf{U}}\|_{\boldsymbol{L}} = \boldsymbol{\chi}_{\mathsf{U}}^* \boldsymbol{L} \boldsymbol{\chi}_{\mathsf{U}} \quad \text{where} \quad \boldsymbol{\chi}_{\mathsf{U}}(u) = \begin{cases} 1, & u \in \mathsf{U}; \\ 0, & u \notin \mathsf{U}. \end{cases}$$

This formula allows us to use algebra to study combinatorial problems.

4.3 Harmonic analysis on multigraphs

We are now prepared to introduce the basic theory of harmonic functions on graphs.

4.3.1 Harmonic functions

Harmonic functions arise as the solutions to homogeneous linear equations involving the Laplacian matrix.

Definition 4.3 (Harmonic function). Let G be a multigraph, and let $\mathsf{U} \subseteq \mathsf{V}$ be a subset of the vertices. A function $\boldsymbol{\varphi} : \mathsf{V} \to \mathbb{R}$ is *harmonic* on U if

$$(\boldsymbol{L}\boldsymbol{\varphi})(u) = 0 \quad \text{for each } u \in \mathsf{U}.$$

In particular, we say that the function $\boldsymbol{\varphi}$ is harmonic at a vertex u if $(\boldsymbol{L}\boldsymbol{\varphi})(u) = 0$.

Our first result provides more intuition: A function $\boldsymbol{\varphi}$ is harmonic at a vertex u when the value $\varphi(u)$ is the weighted average of the values $\varphi(v)$ at its neighbors $v \in \mathsf{N}(u)$.

Proposition 4.4 (Averaging property). The function $\boldsymbol{\varphi} : \mathsf{V} \to \mathbb{R}$ is harmonic at a vertex $u \in \mathsf{V}$ if and only if

$$\varphi(u) = \frac{1}{w(u)} \sum\nolimits_{e=uv} w(e)\, \varphi(v). \tag{4.2}$$

Each distinct multiedge of the form $e = uv$ appears once in the sum!

Proof. This statement follows immediately from the definition (4.1) of the Laplacian and the definition (4.2) of harmonicity. ∎

4.3.2 Example: Hitting probabilities

Let $\mathsf{B} \subseteq \mathsf{V}$ be a distinguished set of vertices. For a starting point $u \in \mathsf{V}$ and a vertex $b \in \mathsf{B}$, the *hitting probability* $h_b(u)$ is the probability that a random walk with initial vertex $u_0 = u$ arrives at b before it arrives at any other vertex of B. Note that

$$h_b(b) = 1 \quad \text{and} \quad h_b(a) = 0 \quad \text{for each } a \in \mathsf{B} \setminus \{b\}. \tag{4.3}$$

For each remaining vertex $u \notin \mathsf{B}$, the hitting probability satisfies a simple recursion:

$$\begin{aligned} h_b(u) &= \sum\nolimits_{v \in \mathsf{N}(u)} \mathbb{P}\{u_1 = v \mid u_0 = u\} \cdot h_b(v) \\ &= \frac{1}{w(u)} \sum\nolimits_{e=uv} w(e)\, h_b(v). \end{aligned} \tag{4.4}$$

Proposition 4.4 now implies that the hitting probability $\boldsymbol{h}_b$ is harmonic on $\mathsf{V} \setminus \mathsf{B}$.

4.3.3 The maximum principle

The averaging property in Proposition 4.4 has a very significant consequence.

> **Theorem 4.5 (Maximum principle).** Let G be a connected multigraph. If $\boldsymbol{\varphi} : \mathsf{V} \to \mathbb{R}$ is harmonic on V, then $\boldsymbol{\varphi}$ is a constant function.

Proof. Suppose that φ is not constant. Introduce the set M of vertices where φ achieves its maximum value:

$$\mathsf{M} = \arg\max\{\varphi(u) : u \in \mathsf{V}\}.$$

Since φ is not constant, M is a proper subset of V. Moreover, since G is connected, we can extract adjacent vertices $u \sim u'$ where $u \in \mathsf{M}$ and $u' \notin \mathsf{M}$. We calculate that

$$\sum\nolimits_{e=uv} w(e)\,\varphi(v) < \left[\sum\nolimits_{e=uv} w(e)\right] \cdot \max_{v \in \mathsf{N}(u)} \varphi(v) = w(u) \cdot \varphi(u).$$

Indeed, there is a multiedge uu' that participates in the sum, and $\varphi(u') < \max\{\varphi(v) : v \in \mathsf{N}(u)\} = \varphi(u)$. Equivalently,

$$\varphi(u) > \frac{1}{w(u)} \sum\nolimits_{e=uv} w(e)\,\varphi(v).$$

Therefore, φ is not harmonic at u. We reject the hypothesis that φ is constant. ∎

4.3.4 Poles

Let us explain why Theorem 4.5 is called a maximum principle.

Definition 4.6 (Pole). Let $\varphi : \mathsf{V} \to \mathbb{R}$ be a function. A vertex $v \in \mathsf{V}$ is called a *pole* of the function if φ is *not* harmonic at v.

Corollary 4.7 (Existence of poles). Let G be a connected multigraph. If $\varphi : \mathsf{V} \to \mathbb{R}$ is a nonconstant function, then φ attains its maximum and minimum value at poles. In particular, φ has at least two poles.

Proof. In the proof of Theorem 4.5, we defined the set M of vertices where a function φ achieves its maximum value. We proved that M contains a vertex u where φ is not harmonic. Therefore, the function φ has a pole, and the maximum occurs there.

We can apply the same argument to the negation $-\varphi$ to identify a pole u' where φ achieves its minimum.

Since φ is not constant, the maximum and minimum are not achieved at the same location. Thus $u \neq u'$. We conclude that φ has at least two poles. ∎

4.3.5 Harmonic extensions

Next, let us consider what happens if we require a harmonic function to meet some boundary conditions.

Definition 4.8 (Harmonic extension). Let $\varphi_0 : \mathsf{B} \to \mathbb{R}$ be a function on a nonempty set $\mathsf{B} \subseteq \mathsf{V}$ of vertices. A *harmonic extension* of φ_0 is a function $\varphi : \mathsf{V} \to \mathbb{R}$ that solves the linear system

$$\begin{cases} (\boldsymbol{L}\varphi)(u) = 0, & u \in \mathsf{V} \setminus \mathsf{B}; \\ \varphi(u) = \varphi_0(u), & u \in \mathsf{B}. \end{cases}$$

We can construct a unique harmonic extension under minimal hypotheses.

Theorem 4.9 (Harmonic extensions). Let G be a connected multigraph. Distinguish a nonempty set $\mathsf{B} \subseteq \mathsf{V}$ of vertices. For any boundary data $\boldsymbol{\varphi}_0 : \mathsf{B} \to \mathbb{R}$, there is a unique harmonic extension of $\boldsymbol{\varphi}_0$ to the set $\mathsf{V} \setminus \mathsf{B}$ of remaining vertices.

Proof. **Uniqueness:** Let $\boldsymbol{\varphi}_1$ and $\boldsymbol{\varphi}_2$ be two harmonic extensions of $\boldsymbol{\varphi}_0$. Consider their difference $\boldsymbol{\psi} = \boldsymbol{\varphi}_1 - \boldsymbol{\varphi}_2$. By linearity, $\boldsymbol{\varphi}$ is harmonic on $\mathsf{V} \setminus \mathsf{B}$:

$$(\boldsymbol{L}\boldsymbol{\psi})(u) = (\boldsymbol{L}\boldsymbol{\varphi}_1)(u) - (\boldsymbol{L}\boldsymbol{\varphi}_2)(u) = 0 \quad \text{for } u \in \mathsf{V} \setminus \mathsf{B}.$$

Corollary 4.7 implies that $\boldsymbol{\psi}$ achieves its maximum and minimum on B. But $\boldsymbol{\psi}$ has zero boundary data:

$$\psi(v) = \varphi_1(v) - \varphi_2(v) = 0 \quad \text{for each } v \in \mathsf{B}.$$

Therefore, $\boldsymbol{\psi}$ is identically equal to zero.

Existence: Using the hitting probabilities (Section 4.3.2), we define the real-valued function

$$\varphi(u) = \sum\nolimits_{b \in \mathsf{B}} \varphi_0(b)\, h_b(u) \quad \text{for each } u \in \mathsf{V}.$$

By the property (4.3) of the hitting probability $\boldsymbol{h}_b$, the function $\boldsymbol{\varphi}$ agrees with $\boldsymbol{\varphi}_0$ on B. Meanwhile, for $u \in \mathsf{V} \setminus \mathsf{B}$, the recursion (4.4) gives

$$\begin{aligned} \varphi(u) &= \sum\nolimits_{b \in \mathsf{B}} \varphi_0(b)\, \frac{1}{w(u)} \sum\nolimits_{e=uv} w(e)\, h_b(v) \\ &= \frac{1}{w(u)} \sum\nolimits_{e=uv} w(e) \sum\nolimits_{b \in \mathsf{B}} \varphi_0(b)\, h_b(v) \\ &= \frac{1}{w(u)} \sum\nolimits_{e=uv} w(e)\, \varphi(v). \end{aligned}$$

Therefore, $\boldsymbol{\varphi}$ is a harmonic extension of $\boldsymbol{\varphi}_0$ from B to V. ∎

4.3.6 Interpretation: Plumbing

Let $\boldsymbol{p} \in \mathbb{R}^{\mathsf{V}}$ denote the pressure at each fixture in a network of pipes. Suppose that the network contains an inlet u_{in} where the pressure $p_0(u_{\text{in}}) > 0$, usually called a *source*. Suppose that the network also contains at outlet u_{out} where the pressure $p_0(u_{\text{out}}) < 0$, usually called a *sink*. The other fixtures are called *internal nodes*.

The theory of hydrodynamics states that the total (signed) flow $f(u)$ at an internal node u equals zero because any water that enters must also leave. Each pipe incident on u contributes to the flow in or out of the fixture u. The rate of flow along a pipe $e = uv$ is (proportional to) the size $w(e)$ of the pipe times the difference $p(u) - p(v)$ in pressure at the endpoints. Altogether,

$$0 = f(u) = \sum\nolimits_{e=uv} w(e)\,(p(u) - p(v)) \quad \text{for each internal } u \in \mathsf{V}.$$

We can rewrite this equation as

$$p(u) = \frac{1}{w(u)} \sum\nolimits_{e=uv} w(e)\, p(v) \quad \text{for each internal } u \in \mathsf{V}.$$

In other words, the pressure $\boldsymbol{p}$ is harmonic at each internal node.

These equations can be combined:

$$\begin{cases} (\boldsymbol{L}\boldsymbol{p})(u) = 0, & u \text{ is internal;} \\ p(u) = p_0(u), & u \in \{u_{\text{in}}, u_{\text{out}}\}. \end{cases}$$

In summary, the pressure $\boldsymbol{p} \in \mathbb{R}^{\mathsf{V}}$ is the harmonic extension of the pressure at the source and sink. If there are many sources and sinks, a similar formula is valid.

4.3.7 Interpretation: Resistor networks

Let $\boldsymbol{\varphi} \in \mathbb{R}^{\mathsf{V}}$ denote the voltage at each node in an electrical network. Suppose that the network contains a source u_{in} where the voltage $\varphi_0(u_{\text{in}}) > 0$; for example, a battery. Suppose that the network also contains a sink u_{out} where the voltage $\varphi_0(u_{\text{out}}) < 0$; for example, the ground. The other fixtures are called *internal nodes*.

The theory of resistor networks states that the total current $f(u)$ flowing through an internal node u equals zero because there is no input or output. Each wire incident on u contributes to the current flowing in or out of the node u. The amount of current flowing along a wire $e = uv$ is proportional to the conductance $w(e)$ and the difference $\varphi(u) - \varphi(v)$ in voltage at the endpoints (i.e., the difference in electrical potential). Altogether,

$$0 = f(u) = \sum\nolimits_{e=uv} w(e)\,(\varphi(u) - \varphi(v)) \quad \text{for each internal } u \in \mathsf{V}.$$

We can rewrite this equation as

$$\varphi(u) = \frac{1}{w(u)} \sum\nolimits_{e=uv} w(e)\,\varphi(v) \quad \text{for each internal } u \in \mathsf{V}.$$

In other words, the voltage $\boldsymbol{\varphi}$ is harmonic at each internal node.

These equations can be combined:

$$\begin{cases} (\boldsymbol{L}\boldsymbol{\varphi})(u) = 0, & u \text{ is internal;} \\ \varphi(u) = \varphi_0(u), & u \in \{u_{\text{in}}, u_{\text{out}}\}. \end{cases}$$

In summary, the voltage $\boldsymbol{\varphi} \in \mathbb{R}^{\mathsf{V}}$ is the harmonic extension of the voltages at the source and sink. If there are many sources and sinks, a similar formula remains valid.

Conversely, we can consider a vector $\boldsymbol{f} \in \mathbb{R}^{\mathsf{V}}$ of external currents. The value $f(u)$ is the amount of current entering (or leaving) the network at vertex u. The network cannot hold current, so we must assume that $\mathbf{1}^*\boldsymbol{f} = \mathbf{0}$. That is, any current that enters must also leave. Then the induced voltages $\boldsymbol{\varphi} \in \mathbb{R}^{\mathsf{V}}$ at each node satisfy

$$\boldsymbol{\varphi} = \boldsymbol{L}^{\dagger}\boldsymbol{f}.$$

One can easily verify that $\boldsymbol{\varphi}$ is harmonic, except at the nodes u where $\boldsymbol{f}(u) \neq 0$. In addition, the total induced voltage $\mathbf{1}^*\boldsymbol{\varphi} = 0$, which reflects the fact that only voltage differences between terminals play a role in determining the flow.

5. Effective Resistance

"Old radio resistors," Wikipedia

This lecture is based on Dan Spielman's Fall 2018 course on spectral graph theory [Spi].

The parallel between harmonic analysis on graphs and electrical networks suggests further analogies. In this lecture, we explore several important concepts that arise from this perspective. We first discuss the notion of the effective resistance between two vertices in a graph. Then we introduce the leverage of an edge, which is a reflection of its importance in determining the graph structure. Using these concepts, we demonstrate that every graph Laplacian can be approximated strongly by the Laplacian of a sparse graph. We realize this approximation by nonuniform randomized sampling.

5.1 Resistance distance

We have introduced the machinery of harmonic functions so that we can understand properties of the pseudoinverse of a Laplacian. Of course, the pseudoinverse plays a role in the solution of linear systems. But it also has interesting physical interpretations related to the properties of the electrical network determined by the graph.

5.1.1 Effective resistance

We begin with an important definition.

Definition 5.1 (Effective resistance). Let G be a connected multigraph on a vertex set V and with Laplacian matrix $\boldsymbol{L}$. For vertices $u, v \in \mathsf{V}$, not necessarily distinct, the *effective resistance* $\varrho(u, v)$ between the vertices u and v is the nonnegative number

$$\varrho(u, v) = (\boldsymbol{\delta}_u - \boldsymbol{\delta}_v)^* \boldsymbol{L}^\dagger (\boldsymbol{\delta}_u - \boldsymbol{\delta}_v).$$

As usual, $\boldsymbol{\delta}_u$ is a standard basis vector, and † denotes the pseudoinverse.

To understand why this quantity is called the effective resistance, note that

$$\boldsymbol{\varphi} = \boldsymbol{L}^\dagger(\boldsymbol{\delta}_u - \boldsymbol{\delta}_v) \in \mathbb{R}^{\mathsf{V}}$$

is the vector of induced voltages if we inject one unit of current at vertex u and extract one unit of current at vertex v. Then

$$\varrho(u, v) = (\boldsymbol{\delta}_u - \boldsymbol{\delta}_v)^* \boldsymbol{\varphi} = \varphi(u) - \varphi(v).$$

In other words, $\varrho(u, v)$ is the voltage difference between the vertices u and v, per unit of current. In other words, we can interpret it as the resistivity of the entire network against passing one unit of current from u to v.

In the hydraulic analogy, we can think about injecting a unit-rate flow at the inlet u and extracting it at the outlet v. The whole plumbing network behaves like a pipe that shunts the fluid between these two fixtures. The number $\varrho(u, v)$ reflects the effective "size" of this compound pipe.

5.1.2 Effective resistance is a metric

A wonderful fact is that the effective resistance induces a metric on the vertex set of a multigraph. This result is an easy consequence of the maximum principle for harmonic functions. It will play a central role in the algorithm for solving Laplacian systems.

Theorem 5.2 (Effective resistance is a metric). Let G be a connected multigraph on the vertex set V. The effective resistance ϱ determines a metric on the vertices. More precisely, for all vertices $t, u, v \in \mathsf{V}$, it holds that

1. $\varrho(u, v) = 0$ if and only if $u = v$.
2. $\varrho(u, v) = \varrho(v, u)$.
3. $\varrho(t, v) \leq \varrho(t, u) + \varrho(u, v)$.

Proof. Let $\boldsymbol{L}$ be the Laplacian of the multigraph G. The first two properties are easy exercises. For the triangle inequality, we define the functions

$$\begin{aligned}
\boldsymbol{\varphi}_{tu} &= \boldsymbol{L}^\dagger(\boldsymbol{\delta}_t - \boldsymbol{\delta}_u), \quad \text{harmonic on } \mathsf{V} \setminus \{t, u\}; \\
\boldsymbol{\varphi}_{uv} &= \boldsymbol{L}^\dagger(\boldsymbol{\delta}_u - \boldsymbol{\delta}_v), \quad \text{harmonic on } \mathsf{V} \setminus \{u, v\}; \\
\boldsymbol{\varphi}_{tv} &= \boldsymbol{L}^\dagger(\boldsymbol{\delta}_t - \boldsymbol{\delta}_v).
\end{aligned}$$

By linearity, these functions are related as $\boldsymbol{\varphi}_{tv} = \boldsymbol{\varphi}_{tu} + \boldsymbol{\varphi}_{uv}$. Taking the inner product of this identity with $\boldsymbol{\delta}_t - \boldsymbol{\delta}_v$ gives

$$\varrho(t, v) = (\boldsymbol{\delta}_t - \boldsymbol{\delta}_v)^* \boldsymbol{\varphi}_{tv} = (\boldsymbol{\delta}_t - \boldsymbol{\delta}_v)^* \boldsymbol{\varphi}_{tu} + (\boldsymbol{\delta}_t - \boldsymbol{\delta}_v)^* \boldsymbol{\varphi}_{uv}.$$

It remains to bound the right-hand side in terms of the effective resistances $\varrho(t, u)$ and $\varrho(u, v)$. We can accomplish this via the maximum principle.

To that end, we note the relation

$$\varphi_{tu}(t) - \varphi_{tu}(u) = \varrho(t, u) \geq 0.$$

By the maximum principle (Corollary 4.7), the harmonic function $\boldsymbol{\varphi}_{tu}$ takes its maximum value at the pole t and its minimum at the pole u. Thus,

$$\begin{aligned}(\boldsymbol{\delta}_t - \boldsymbol{\delta}_v)^* \boldsymbol{\varphi}_{tu} &= \varphi_{tu}(t) - \varphi_{tu}(v) \\ &\le \varphi_{tu}(t) - \varphi_{tu}(u) = (\boldsymbol{\delta}_t - \boldsymbol{\delta}_u)^* \boldsymbol{\varphi}_{tu} = \varrho(t, u).\end{aligned}$$

Similarly,

$$\begin{aligned}(\boldsymbol{\delta}_t - \boldsymbol{\delta}_v)^* \boldsymbol{\varphi}_{uv} &= \varphi_{uv}(t) - \varphi_{uv}(v) \\ &\le \varphi_{uv}(u) - \varphi_{uv}(v) = (\boldsymbol{\delta}_u - \boldsymbol{\delta}_v)^* \boldsymbol{\varphi}_{uv} = \varrho(u, v).\end{aligned}$$

The result follows when we sequence the last three displays. ∎

5.1.3 An alternative representation

There is another way of writing the effective resistance that will be useful for us. Let us introduce another piece of notation.

Definition 5.3 (Normalizing map). Let G be a connected multigraph with Laplacian matrix $\boldsymbol{L}$. Define the *normalizing map*

$$\boldsymbol{\Phi}(\boldsymbol{M}) = \boldsymbol{\Phi}_{\mathsf{G}}(\boldsymbol{M}) = (\boldsymbol{L}^\dagger)^{1/2} \boldsymbol{M} (\boldsymbol{L}^\dagger)^{1/2} \quad \text{for } \boldsymbol{M} \in \mathbb{H}_{\mathsf{V}}.$$

The exponent $^{1/2}$ extracts the unique psd square root. The normalizing map $\boldsymbol{\Phi}$ is associated with the Laplacian of a particular multigraph G, which will remain fixed throughout our discussion.

Let us note some properties of this map. First, $\boldsymbol{\Phi}(\boldsymbol{L}) = \boldsymbol{P}$, where $\boldsymbol{P}$ is the orthogonal projector onto range$(\boldsymbol{L})$. The function $\boldsymbol{\Phi}$ is an example of a positive linear map. Among many other properties,

$$\boldsymbol{M} \succcurlyeq \mathbf{0} \quad \text{implies} \quad \boldsymbol{\Phi}(\boldsymbol{M}) \succcurlyeq \mathbf{0}.$$

See the book [Bha07] for an introduction to the theory of positive linear maps.

The normalizing map gives us another mechanism for expressing the effective resistance between two vertices. Indeed, the effective resistance is the spectral norm of the normalized elementary Laplacian of the unit edge connecting the two vertices.

Proposition 5.4 (Effective resistance). Let G be a connected multigraph on the vertex set V and with normalizing map $\boldsymbol{\Phi}$. For vertices $u, v \in \mathsf{V}$,

$$\varrho(u, v) = \|\boldsymbol{\Phi}(\boldsymbol{\Delta}_{uv})\|.$$

As always, $\|\cdot\|$ is the spectral norm.

Proof. Since the effective resistance is nonnegative,

$$\begin{aligned}\varrho(u, v) &= \|(\boldsymbol{\delta}_u - \boldsymbol{\delta}_v)^* (\boldsymbol{L}^\dagger)^{1/2} (\boldsymbol{L}^\dagger)^{1/2} (\boldsymbol{\delta}_u - \boldsymbol{\delta}_v)\| \\ &= \|(\boldsymbol{L}^\dagger)^{1/2} (\boldsymbol{\delta}_u - \boldsymbol{\delta}_v)(\boldsymbol{\delta}_u - \boldsymbol{\delta}_v)^* (\boldsymbol{L}^\dagger)^{1/2}\| = \|(\boldsymbol{L}^\dagger)^{1/2} \boldsymbol{\Delta}_{uv} (\boldsymbol{L}^\dagger)^{1/2}\|.\end{aligned}$$

We make the transition to the second line using the relation $\|\boldsymbol{M}\boldsymbol{M}^*\| = \|\boldsymbol{M}^*\boldsymbol{M}\|$. Identify the normalizing map to complete the argument. ∎

Exercise 5.1 Prove that $\varrho(u, v) = \operatorname{tr} \boldsymbol{\Phi}(\boldsymbol{\Delta}_{uv})$.

5.1.4 Leverage of a multiedge

We are now prepared to introduce a notion of the importance of a multiedge to the graph structure.

Definition 5.5 (Leverage). Let G be a connected multigraph on a vertex set V and with normalizing map $\mathbf{\Phi}$. For each multiedge $e = uv$ with weight $w(e)$, the *leverage* of the multiedge e is the quantity

$$\ell(e) = w(e)\, \varrho(u, v) = w(e)\, \|\mathbf{\Phi}(\mathbf{\Delta}_e)\|.$$

As usual, ϱ is the effective resistance induced by the multigraph G.

Proposition 5.6 (Leverage of a multiedge). Let G be a connected multigraph. For each multiedge e in the multigraph, the leverage $\ell(e) \leq 1$.

Proof. Introduce the Laplacian $\boldsymbol{L}$ of the multigraph:

$$\boldsymbol{L} = \sum\nolimits_{e \in \mathsf{G}} w(e)\, \mathbf{\Delta}_e.$$

Apply the normalizing map to the last display:

$$\boldsymbol{P} = \mathbf{\Phi}(\boldsymbol{L}) = \sum\nolimits_{e \in \mathsf{G}} w(e)\, \mathbf{\Phi}(\mathbf{\Delta}_e). \tag{5.1}$$

Since $\mathbf{\Phi}$ is a positive linear map,

$$\boldsymbol{P} \succcurlyeq w(e)\, \mathbf{\Phi}(\mathbf{\Delta}_e) \quad \text{for each } e \in \mathsf{G}.$$

Taking the spectral norm, for each multiedge $e = uv \in \mathsf{G}$, we have

$$1 \geq w(e)\, \|\mathbf{\Phi}(\mathbf{\Delta}_e)\| = w(e)\, \varrho(u, v) = \ell(e).$$

The last identity follows from Proposition 5.4. ∎

The basic idea is that the effective resistance $\varrho(u, v)$ measures how much voltage we need to push a unit of current from the node u to the node v. Meanwhile, the weight $w(e)$ is proportional to the conductance of a wire that connects u and v. As a consequence of Ohm's law, the leverage $\ell(e)$ measures the fraction of current that travels along the wire e if we push one unit of current from u to v.

If there is only one way to get from u to v, all the current must pass along the wire $e = uv$, and the leverage equals one. (Think of the edge connecting the two ends of a barbell graph.) Conversely, if there are many ways to get from u to v, some of the current may follow other routes, and the leverage of $e = uv$ can be small. (Think of the edges in a complete graph.)

Similarly, if two vertices are wired in parallel, each of the multiedges will carry an equal proportion of the current between the vertices. By increasing the number of multiedges, we thereby decrease the leverage of each one to an equal part of the total.

Exercise 5.2 (Total leverage). Assume that G is a connected multigraph on n vertices. Prove that the total of all leverage scores is $n - 1$. That is,

$$\sum\nolimits_{e \in \mathsf{G}} \ell(e) = n - 1.$$

Hint: Use the identity (5.1) and Exercise 5.1.

5.2 Approximating a Laplacian by sampling

As an application of these ideas, we will prove that every Laplacian is well-approximated by the Laplacian of a sparse graph. We construct the sparse graph by randomly sampling edges according to their leverage. The result and high-level approach are due to Spielman & Srivastava [SS11]. The main tool in our analysis is the matrix Bernstein inequality, Theorem 1.13.

5.2.1 Spectral approximation

Let $\boldsymbol{L}$ be the Laplacian of a connected graph G on a set V of n vertices, with normalizing map $\boldsymbol{\Phi}$. Let $\boldsymbol{S}$ be the Laplacian of another graph on the same vertex set V. We are interested in a very strong notion of approximation between these two Laplacians.

Definition 5.7 (Spectral approximation). For $\varepsilon \in (0, 1)$, we say that $\boldsymbol{S}$ is a *ε-spectral approximation* of $\boldsymbol{L}$ when

$$(1 - \varepsilon)\,\boldsymbol{L} \preccurlyeq \boldsymbol{S} \preccurlyeq (1 + \varepsilon)\,\boldsymbol{L}.$$

If $\boldsymbol{S}$ is a spectral approximation of $\boldsymbol{L}$, then the two Laplacians represent graphs with similar properties. Among other things,

1. The effective resistance between a pair of vertices in $\boldsymbol{S}$ is comparable to the effective resistance between the same pair of vertices in $\boldsymbol{L}$.
2. The value of every graph cut in $\boldsymbol{S}$ is comparable to the value of the same cut in $\boldsymbol{L}$.
3. The solution to the linear system $\boldsymbol{S}\boldsymbol{x} = \boldsymbol{f}$ is not very different from the solution of $\boldsymbol{L}\boldsymbol{x} = \boldsymbol{f}$.

The last fact will be very important when we talk about how to solve Laplacian linear systems efficiently.

It is convenient to convert the spectral approximation condition into another form that is more amenable to analysis.

Proposition 5.8 (Spectral approximation). Let $\boldsymbol{L}$ be the Laplacian of a connected graph, and let $\boldsymbol{S}$ be the Laplacian of another graph on the same vertex set. Suppose that

$$\|\boldsymbol{\Phi}(\boldsymbol{S} - \boldsymbol{L})\| \leq \varepsilon.$$

Then $\boldsymbol{S}$ is a spectral approximation of $\boldsymbol{L}$ with quality ε.

Proof. First, subtract $\boldsymbol{L}$ from both sides of the relation:

$$-\varepsilon\,\boldsymbol{L} \preccurlyeq \boldsymbol{S} - \boldsymbol{L} \preccurlyeq +\varepsilon\,\boldsymbol{L}.$$

Apply the normalizing map to this relation to obtain the equivalent condition

$$-\varepsilon\,\boldsymbol{P} \preccurlyeq \boldsymbol{\Phi}(\boldsymbol{S} - \boldsymbol{L}) \preccurlyeq +\varepsilon\,\boldsymbol{P}.$$

Since the range of $\boldsymbol{\Phi}(\boldsymbol{S} - \boldsymbol{L})$ is contained in the range of $\boldsymbol{P}$, it is sufficient to prove that

$$\|\boldsymbol{\Phi}(\boldsymbol{S} - \boldsymbol{L})\| \leq \varepsilon \|\boldsymbol{P}\| = \varepsilon.$$

The last step uses the fact that $\boldsymbol{P}$ is an orthogonal projector. ∎

5.2.2 The sampling model

The representation of the graph Laplacian as a sum of weighted edges allows us to construct a matrix approximation by random sampling, similar to what we did in Lecture 2.

Recall that the Laplacian $\boldsymbol{L}$ admits the representation

$$\boldsymbol{L} = \sum\nolimits_{e \in \mathsf{G}} w(e)\, \boldsymbol{\Delta}_e.$$

To construct a sparse Laplacian that approximates $\boldsymbol{L}$, we introduce a random elementary Laplacian:

$$\boldsymbol{R} = \frac{w(e)}{p_e} \boldsymbol{\Delta}_e \quad \text{with probability } p_e > 0 \text{ for each } e \in \mathsf{G}.$$

It is immediate that $\mathbb{E}\, \boldsymbol{R} = \boldsymbol{L}$. For a parameter $K \geq 1$, we construct the Laplacian approximation by averaging K copies of $\boldsymbol{R}$:

$$\boldsymbol{S} = \frac{1}{K} \sum\nolimits_{i=1}^{K} \boldsymbol{R}_i \quad \text{where } \boldsymbol{R}_i \sim \boldsymbol{R} \text{ iid.}$$

Then $\boldsymbol{S}$ is the Laplacian of a weighted graph with at most K edges, and $\boldsymbol{S}$ is an unbiased estimator of $\boldsymbol{L}$.

5.2.3 The sampling probabilities

Our goal is to control the size of $\boldsymbol{\Phi}(\boldsymbol{S} - \boldsymbol{L})$, and we will exploit our analysis of matrix sampling estimators, Theorem 2.1. To activate this result, we need to make each summand uniformly bounded. To that end, calculate that

$$\|\boldsymbol{\Phi}(\boldsymbol{R})\| = \frac{w(e)}{p_e} \|\boldsymbol{\Phi}(\boldsymbol{\Delta}_e)\| = \frac{\ell(e)}{p_e}.$$

Therefore, it is natural to select the sampling probabilities proportional to the leverage of the edges: $p_e = c\ell(e)$. The constant is selected to obtain a probability mass:

$$1 = \sum\nolimits_e p_e = c \sum\nolimits_e \ell(e) = (n-1)c.$$

We have used Exercise 5.2 here. On other words, $c = 1/(n-1)$.

5.2.4 The analysis

The analysis is easy now. The upper bound parameter in Theorem 2.1 satisfies

$$B = \sup \|\boldsymbol{\Phi}(\boldsymbol{R})\| = \sup \frac{\ell(e)}{\ell(e)/(n-1)} = n - 1.$$

The per-sample second moment satisfies

$$\begin{aligned} m_2(\boldsymbol{\Phi}(\boldsymbol{R})) = \|\mathbb{E}\, \boldsymbol{\Phi}(\boldsymbol{R})^2\| &\leq \big\|\mathbb{E}\, \big[\|\boldsymbol{\Phi}(\boldsymbol{R})\| \cdot \boldsymbol{\Phi}(\boldsymbol{R})\big]\big\| \\ &\leq (n-1)\, \|\boldsymbol{\Phi}(\mathbb{E}\, \boldsymbol{R})\| = (n-1)\, \|\boldsymbol{P}\| = n - 1. \end{aligned}$$

Theorem 2.1 immediately implies that

$$\begin{aligned}\mathbb{E}\,\|\boldsymbol{\Phi}(\boldsymbol{S}-\boldsymbol{L})\| &\le \sqrt{\frac{2m_2(\boldsymbol{R})\log(2n)}{K}} + \frac{2B\log(2n)}{3K} \\ &\le \sqrt{\frac{2(n-1)\log(2n)}{K}} + \frac{2(n-1)\log(2n)}{3K}.\end{aligned}$$

Set $K = 4\varepsilon^{-2}(n-1)\log(2n)$ to arrive at the bound

$$\mathbb{E}\,\|\boldsymbol{\Phi}(\boldsymbol{S}-\boldsymbol{L})\| < \varepsilon.$$

The final estimate assumes that $\varepsilon \le 1$.

By the probabilistic method, every graph with sufficiently small leverage scores admits an ε-spectral approximation with at most $4\varepsilon^{-2}n\log n$ edges. Note that this bound is *independent* of the number m of edges in the target graph G!

Let us remark that the approach here tracks the presentation in Lecture 2. We recognize that the representation of the Laplacian as a sum of elementary Laplacians furnishes a breakdown of the matrix into simple components. Once we agree that our goal is to obtain an ε-spectral approximation, the matrix sampling result, Theorem 2.1, tells us exactly what properties the sampling probability ought to have. The leverage emerges as a formal consequence.

5.2.5 Computational aspects

This argument gives an algorithm for constructing a sparse graph that is a spectral approximation of an arbitrary graph. Unfortunately, to implement this method, we need to compute the leverages so that we can perform the sampling. The naïve approach to this problem involves $\Theta(n^3)$ computation. The literature on theoretical algorithms contains techniques that can achieve this goal more efficiently, but these methods may not be entirely practical. See [Spi] for discussion and references.

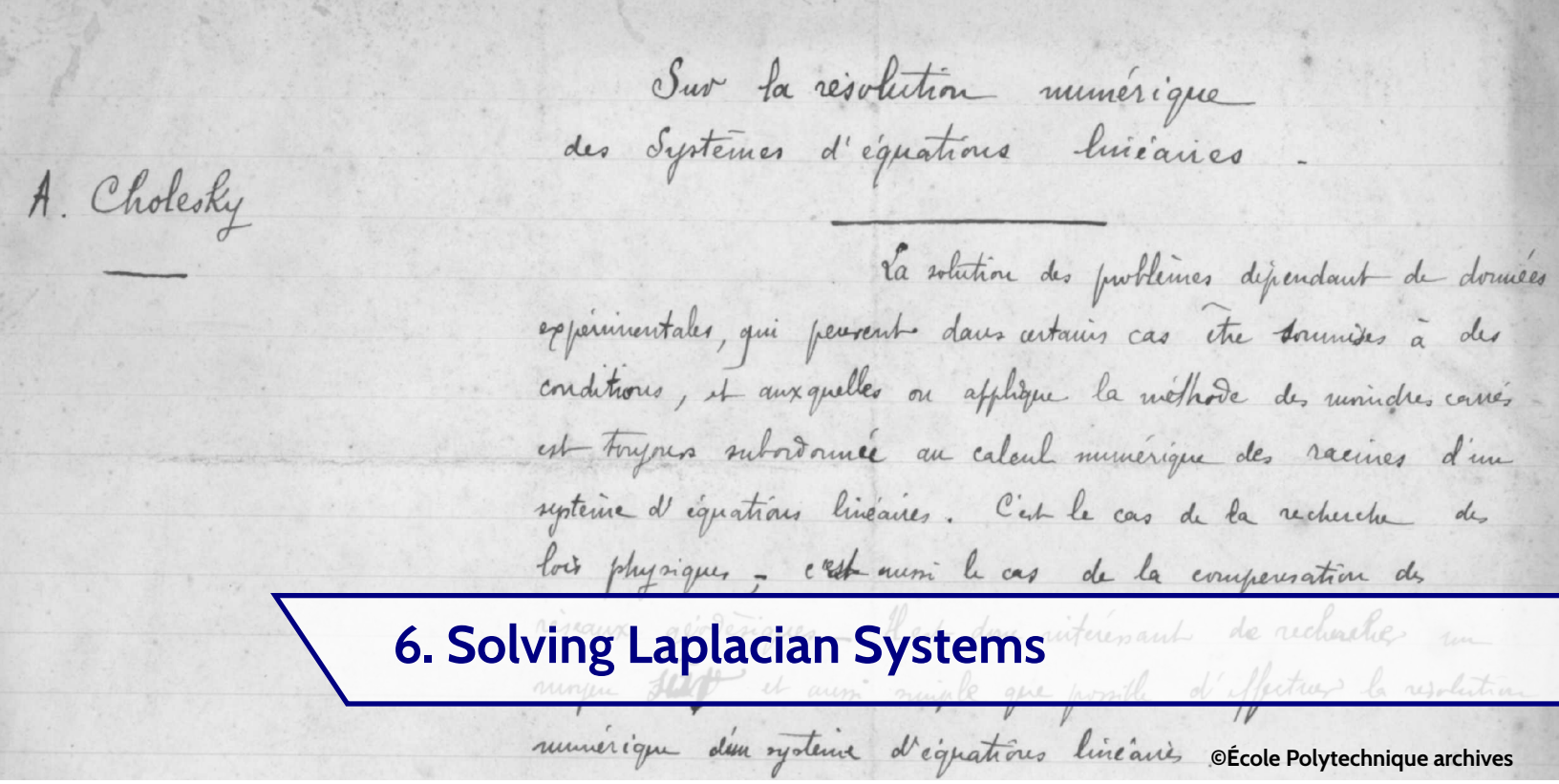
5.2.6 Conclusion

To conclude, we have established the following theorem.

> **Theorem 5.9 (Spielman & Srivastava, 2011).** Let G be a connected graph on n vertices with Laplacian $\boldsymbol{L}$. Fix a parameter $\varepsilon \in (0, 1)$. Then there is a connected graph on the same vertex set, with at most $4\varepsilon^{-2}n\log n$ edges, and whose Laplacian is an ε-spectral approximation to $\boldsymbol{L}$.

The analysis of this sampling estimator cannot be improved in general beyond the constants. Indeed, for a complete graph, the leverage of each edge is the same. The sampling technique chooses each edge with equal probability, so an individual sample is equally likely to be incident on each vertex. But the coupon collector problem tells us that we need $\Theta(n\log n)$ samples to acquire edges incident on all n vertices. This outcome is prerequisite for $\boldsymbol{S}$ to be the Laplacian of a connected graph, which is necessary if $\boldsymbol{S}$ is to approximate $\boldsymbol{L}$ spectrally.

Nevertheless, the sparsity bound in the theorem is not optimal for graph approximations. See [BSS14] for a sharp result based on a deterministic construction.



©École Polytechnique archives

6. Solving Laplacian Systems

This lecture is adapted from Rasmus Kyng's dissertation [Kyn17].

In this lecture, we will discuss computational methods for solving Laplacian linear systems. The classic direct method is based on computing the Cholesky decomposition of the Laplacian matrix. This decomposition takes a special form for the Laplacian, as compared with a general psd matrix. This special form will serve as the foundation for developing a very efficient algorithm for solving graph Laplacian systems, as we will see in Lecture 8.

6.1 Cholesky meets Laplace

This section gives an overview of a classic approach for solving a Laplacian system.

6.1.1 Setup

Let G be a connected multigraph. The vertex set $\mathsf{V} = \{1, \ldots, n\}$. The multiedge set E comprises m edges. The weight function $w_{\mathsf{G}} : \mathsf{E} \to \mathbb{R}_{++}$. As always, $\boldsymbol{L}$ denotes the weighted Laplacian matrix of the graph. We will treat the multigraph and the Laplacian as interchangeable by presenting the Laplacian as a weighted sum of multiedges.

6.1.2 Laplacian systems

Suppose that we are given a forcing vector $\boldsymbol{f} \in \mathbb{R}^{\mathsf{V}}$ that is orthogonal to the constant vector $\mathbf{1}^* \boldsymbol{f} = 0$. Our aim is to solve the linear system

$$\boldsymbol{L}\boldsymbol{x} = \boldsymbol{f}. \tag{6.1}$$

Write $\boldsymbol{x}_\star \in \mathbb{R}^{\mathsf{V}}$ for the (unique) solution to this system with $\mathbf{1}^* \boldsymbol{x}_\star = 0$.

6.1.3 Solution via Cholesky decomposition

A standard approach to solving a Laplacian linear system is to extract a Cholesky decomposition of the Laplacian:

$$\boldsymbol{L} = \boldsymbol{C}\boldsymbol{C}^* \quad \text{where } \boldsymbol{C} \text{ is lower-triangular.}$$

We will spend most of this lecture going over the details of how Cholesky decomposition works for Laplacian matrices. The cost of producing this decomposition is usually $\Theta(n^3)$ arithmetic operations. Owing to fill-in, the lower-triangular factor $\boldsymbol{C}$ often has fully $\Theta(n^2)$ nonzero entries.

Once we have the triangular factorization, we can solve the linear system (6.1) with $\Theta(n^2)$ arithmetic operations. Indeed,

$$\boldsymbol{x}_\star = \boldsymbol{L}^\dagger \boldsymbol{f} = (\boldsymbol{C}^*)^\dagger (\boldsymbol{C}^\dagger \boldsymbol{f}).$$

We can apply $\boldsymbol{C}^\dagger$ using forward substitution in time $\Theta(n^2)$, and we can apply $(\boldsymbol{C}^*)^\dagger$ using backward substitution in time $\Theta(n^2)$. This approach produces results that are accurate (almost) to machine precision [Hig02, Chaps. 8, 10].

6.2 Cholesky decomposition: Matrix view

Let us explain the process of computing the Cholesky decomposition of a psd matrix. We begin with a linear-algebraic treatment that is applicable to any matrix. In the next section, we specialize to the case of Laplacian matrices.

6.2.1 Setup

Let $\boldsymbol{M} \in \mathbb{H}_n$ be a psd matrix. The Cholesky decomposition iteratively reduces the psd matrix to a product of lower-triangular factors.

6.2.2 First step of the Cholesky decomposition

Let us begin with a visual description of the first step in the Cholesky process. Writing out the first row and column explicitly, we can express the matrix $\boldsymbol{M}$ as

$$\boldsymbol{M} = \begin{bmatrix} d & -\boldsymbol{a}^* \\ -\boldsymbol{a} & \boldsymbol{M}_2 \end{bmatrix}.$$

In this expression, d is a nonnegative number (because $\boldsymbol{M}$ is psd), $\boldsymbol{a} \in \mathbb{R}^{n-1}$ and $\boldsymbol{M}_2 \in \mathbb{H}_{n-1}$. Construct the rank-one psd matrix

$$d^\dagger \begin{bmatrix} d \\ -\boldsymbol{a} \end{bmatrix} \begin{bmatrix} d \\ -\boldsymbol{a} \end{bmatrix}^* = \begin{bmatrix} d & -\boldsymbol{a}^* \\ -\boldsymbol{a} & d^\dagger \boldsymbol{a}\boldsymbol{a}^* \end{bmatrix}.$$

(Since $\boldsymbol{M}$ is psd, if the diagonal entry $d = 0$, then also $\boldsymbol{a} = \boldsymbol{0}$.) Therefore, we can eliminate the first row and column of $\boldsymbol{M}$ by subtracting this rank-one matrix:

$$\boldsymbol{M}/1 = \boldsymbol{M} - d^\dagger \begin{bmatrix} d \\ -\boldsymbol{a} \end{bmatrix} \begin{bmatrix} d \\ -\boldsymbol{a} \end{bmatrix}^* = \begin{bmatrix} 0 & \boldsymbol{0}^* \\ \boldsymbol{0} & \boldsymbol{M}_2 - d^\dagger \boldsymbol{a}\boldsymbol{a}^* \end{bmatrix}.$$

The notation $\boldsymbol{M}/1$ refers to the Schur complement of $\boldsymbol{M}$ with respect to the first coordinate subspace.

Observe that the reduced matrix $\boldsymbol{M}/1$ remains psd. Indeed, for each vector $\boldsymbol{x} \in \mathbb{R}^{n-1}$, define $\alpha = d^{\dagger}\boldsymbol{a}^*\boldsymbol{x}$, and calculate that

$$0 \le \begin{bmatrix} \alpha \\ \boldsymbol{x} \end{bmatrix}^* \boldsymbol{M} \begin{bmatrix} \alpha \\ \boldsymbol{x} \end{bmatrix} = d\alpha^2 - 2\alpha \boldsymbol{a}^*\boldsymbol{x} + \boldsymbol{x}^*\boldsymbol{M}_2\boldsymbol{x} = \boldsymbol{x}^*(\boldsymbol{M}_2 - d^{\dagger}\boldsymbol{a}\boldsymbol{a}^*)\boldsymbol{x}.$$

Since $\boldsymbol{x}$ is arbitrary, the Schur complement $\boldsymbol{M}/1$ is psd.

We can apply this elimination procedure to each remaining coordinate in sequence, reducing the size of the nonzero block at each step.

6.2.3 Cholesky decomposition, without pivoting

Here is a more formal description of the Cholesky decomposition, where we eliminate the coordinates in lexicographic order.

To begin the process, set $\boldsymbol{S}_0 = \boldsymbol{M}$. At each step $i = 1, 2, \dots, n$, we eliminate the ith coordinate. We write $u_i = i$ to emphasize the difference between the choice of coordinate (u_i) and the step (i) in the iteration procedure. Introduce the vector

$$\boldsymbol{c}_i = \frac{1}{\sqrt{(\boldsymbol{S}_{i-1})_{u_i u_i}}} \cdot \boldsymbol{S}_{i-1}\boldsymbol{\delta}_{u_i}.$$

(If the diagonal entry $(\boldsymbol{S}_{i-1})_{u_i u_i}$ happens to equal zero, we set $\boldsymbol{c}_i = \boldsymbol{0}$.) Zero out the row and column in $\boldsymbol{S}_{i-1}$ indexed by the coordinate u_i by forming the Schur complement:

$$\boldsymbol{S}_i = \boldsymbol{S}_{i-1}/u_i = \boldsymbol{S}_{i-1} - \boldsymbol{c}_i\boldsymbol{c}_i^*.$$

We continue this process for n steps.

At each iteration i, the matrix $\boldsymbol{S}_{i-1}$ is psd, and it has the block form

$$\boldsymbol{S}_{i-1} = \begin{bmatrix} \boldsymbol{0}_{i\times i} & \boldsymbol{0}_{i\times(n-i)} \\ \boldsymbol{0}_{(n-i)\times i} & \star \end{bmatrix}.$$

The symbol $\star$ indicates an $(n-i)\times(n-i)$ block of nonzero coordinates. In particular, after n steps of this procedure, $\boldsymbol{S}_n = \boldsymbol{0}$.

When the iteration is complete, we collect the vectors $\boldsymbol{c}_i$ into a matrix:

$$\boldsymbol{C} = \begin{bmatrix} \boldsymbol{c}_1 & \boldsymbol{c}_2 & \dots & \boldsymbol{c}_n \end{bmatrix} \in \mathbb{R}^{n\times n}.$$

Since $\boldsymbol{S}_{i-1}$ is supported on the coordinates $\{i, \dots, n\}$, so is the vector $\boldsymbol{c}_i$. Therefore, the matrix $\boldsymbol{C}$ is lower-triangular.

To understand the role of the matrix $\boldsymbol{C}$, observe that

$$\boldsymbol{M} = \boldsymbol{S}_0 - \boldsymbol{S}_n = \sum\nolimits_{i=1}^{n} (\boldsymbol{S}_{i-1} - \boldsymbol{S}_i) = \sum\nolimits_{i=1}^{n} \boldsymbol{c}_i\boldsymbol{c}_i^* = \boldsymbol{C}\boldsymbol{C}^*.$$

The factorization $\boldsymbol{M} = \boldsymbol{C}\boldsymbol{C}^*$ is called a Cholesky decomposition of the input matrix $\boldsymbol{M}$. The Cholesky procedure progressively decomposes the input matrix:

$$\boldsymbol{M} = \boldsymbol{S}_i + \sum\nolimits_{k=1}^{i} \boldsymbol{c}_k\boldsymbol{c}_k^* \quad \text{for each } i = 0, 1, 2, \dots, n.$$

This relation breaks down the process into the part of the matrix that remains to be factored ($\boldsymbol{S}_i$) and the part of the factorization that is done (the sum of rank-one terms).

6.2.4 Cholesky decomposition, with pivoting

It is not necessary to eliminate the coordinates in the lexicographic order. At iteration i, suppose instead that we eliminate the coordinate $u_i = \pi(i)$, where $\pi : \{1, \dots, n\} \to \{1, \dots, n\}$ is a permutation (i.e., a bijection). The permutation π can be chosen in advance, or $\pi(i)$ may be selected at iteration i. Otherwise, the decomposition algorithm is the same as before.

In this case, the computed matrix $\boldsymbol{C}$ is morally lower-triangular. The permutation π gives the order of elimination for solving the system $\boldsymbol{C}\boldsymbol{x} = \boldsymbol{f}$ by substitution. We omit further details.

6.2.5 Computational cost

At the ith step of the Cholesky decomposition, the cost of computing the Schur complement is $\Theta((n-i)^2)$ arithmetic operations. Therefore, the total cost of n iterations is $\Theta(n^3)$.

When the input matrix is sparse, we may be able to economize during the early iterations by exploiting sparsity. Nevertheless, each time we form a Schur complement, the matrix often becomes denser, a process known as *fill-in*. It is hard to avoid fill-in, except in special cases. As a consequence, we generally need $\Theta(n^3)$ operations to obtain the Cholesky decomposition. This is very expensive.

6.3 Cholesky decomposition: Graph view

When we apply the Cholesky algorithm to a Laplacian, we can interpret the basic step as a combinatorial operation on a multigraph.

6.3.1 Setup

Let G be a connected multigraph with Laplacian matrix $\boldsymbol{L}$. We can interpret the Cholesky decomposition of the Laplacian matrix $\boldsymbol{L}$ in graph-theoretic terms. In particular, we can express the matrices that arise during the process as Laplacians!

6.3.2 First step of the Cholesky decomposition

To illustrate the idea, suppose that we want to eliminate the first vertex. Isolating the role of the first vertex,

$$\boldsymbol{L} = \sum\nolimits_{e \in \mathsf{G}} w(e)\, \boldsymbol{\Delta}_e = \begin{bmatrix} d & -\boldsymbol{a}^* \\ -\boldsymbol{a} & \boldsymbol{L}_2 \end{bmatrix}.$$

Since $\boldsymbol{L}$ is a Laplacian, we can say more about the terms that appear here:

$$d = w_{\boldsymbol{L}}(1) \geq 0 \quad \text{and} \quad \boldsymbol{a} \geq \boldsymbol{0} \quad \text{and} \quad \boldsymbol{a}^*\boldsymbol{1} = d.$$

Indeed, the first diagonal entry of the Laplacian is the total weight $w_{\boldsymbol{L}}(1)$ of the first vertex. We regard $\boldsymbol{a} \in \mathbb{R}^{\mathsf{V}\setminus\{1\}}$. The off-diagonal entries in the Laplacian are nonpositive, and the number a_v is the total weight of all multiedges of the form $e = 1v$ for each vertex $v \neq 1$. The identity $\boldsymbol{a}^*\boldsymbol{1} = d$ reflects the fact that the diagonal entry $w_{\boldsymbol{L}}(1)$ is

the sum of all the weights of multiedges incident on the first vertex. We write $\boldsymbol{L}_2$ as a placeholder for the submatrix indexed by the vertices $\mathsf{V} \setminus \{1\}$.

To compute the Schur complement $\boldsymbol{L}/1$ with respect to the first vertex, we subtract a rank-one matrix from the Laplacian. Introduce the vector

$$\boldsymbol{c} = \frac{1}{\sqrt{d}} \boldsymbol{L}\boldsymbol{\delta}_1 = \frac{1}{\sqrt{d}} \begin{bmatrix} d \\ -\boldsymbol{a} \end{bmatrix}, \quad \text{so that} \quad \boldsymbol{c}\boldsymbol{c}^* = \begin{bmatrix} d & -\boldsymbol{a}^* \\ -\boldsymbol{a} & d^\dagger \boldsymbol{a}\boldsymbol{a}^* \end{bmatrix}.$$

(If $d = 0$, we interpret the fraction as computing a pseudoinverse.) Then the Schur complement takes the form

$$\boldsymbol{L}/1 = \boldsymbol{L} - \boldsymbol{c}\boldsymbol{c}^* = \begin{bmatrix} 0 & \boldsymbol{0}^* \\ \boldsymbol{0} & \boldsymbol{L}_2 - d^\dagger \boldsymbol{a}\boldsymbol{a}^* \end{bmatrix}.$$

To understand what is really happening here, we will think about this operation as the composition of two steps.

Define the Laplacian of the set of multiedges incident on the first vertex:

$$\textsc{star}(1) = \sum\nolimits_{e=1v} w(e)\, \boldsymbol{\Delta}_e = \begin{bmatrix} d & -\boldsymbol{a}^* \\ -\boldsymbol{a} & \operatorname{diag}(\boldsymbol{a}) \end{bmatrix}.$$

Here, $\operatorname{diag}(\boldsymbol{a})$ is the diagonal matrix determined by the vector $\boldsymbol{a}$. This Laplacian is called the *star* induced by the first vertex. Adding and subtracting $\textsc{star}(1)$ from the Schur complement $\boldsymbol{L}/1$, we obtain

$$\boldsymbol{L}/1 = \big(\boldsymbol{L} - \textsc{star}(1)\big) + \big(\textsc{star}(1) - \boldsymbol{c}\boldsymbol{c}^*\big).$$

We will check that each of the large parentheses defines a Laplacian matrix. Since the class of Laplacians is closed under addition, the Schur complement $\boldsymbol{L}/1$ is also a Laplacian matrix!

The first parenthesis is simply the Laplacian of the multigraph obtained by removing from G the multiedges incident on the first vertex:

$$\boldsymbol{L} - \textsc{star}(1) = \sum\nolimits_{e \not\ni 1} w(e)\, \boldsymbol{\Delta}_e.$$

This point follows immediately from the definitions. Observe that none of the remaining multiedges is incident on the first vertex. This is equivalent to the matrix $\boldsymbol{L} - \textsc{star}(1)$ being supported on the coordinates $v \neq 1$.

Consider the second parenthesis:

$$\textsc{star}(1) - \boldsymbol{c}\boldsymbol{c}^* = \begin{bmatrix} 0 & \boldsymbol{0}^* \\ \boldsymbol{0} & \operatorname{diag}(\boldsymbol{a}) - d^\dagger \boldsymbol{a}\boldsymbol{a}^* \end{bmatrix}.$$

This matrix is also a Laplacian! Indeed, by direct calculation, the diagonal entries are nonnegative, the off-diagonal entries are nonpositive, and each row sums to one. Alternatively, we can write

$$\begin{aligned} \begin{bmatrix} 0 & \boldsymbol{0}^* \\ \boldsymbol{0} & \operatorname{diag}(\boldsymbol{a}) - d^\dagger \boldsymbol{a}\boldsymbol{a}^* \end{bmatrix} &= \frac{1}{2d} \sum\nolimits_{v_1, v_2 \neq 1} a_{v_1} a_{v_2} (\boldsymbol{\delta}_{v_1} - \boldsymbol{\delta}_{v_2})(\boldsymbol{\delta}_{v_1} - \boldsymbol{\delta}_{v_2})^* \\ &= \frac{1}{2w_{\boldsymbol{L}}(1)} \sum_{\substack{e_1 = 1v_1 \\ e_2 = 1v_2}} w(e_1)\, w(e_2)\, \boldsymbol{\Delta}_{v_1 v_2}. \end{aligned}$$

(We interpret the fraction bar as computing a pseudoinverse.) This Laplacian is also called the *clique* induced by eliminating the vertex 1.

To wit, the process of computing the Schur complement of a Laplacian with respect to a vertex amounts to removing the star induced by the vertex and adding back the clique induced by eliminating the vertex.

6.3.3 Stars and cliques

Let us develop this construction in more generality. Let $\boldsymbol{S}$ be the Laplacian of a multigraph on the vertex set V, expressed as a weighted sum of multiedges:

$$\boldsymbol{S} = \sum\nolimits_{e \in \boldsymbol{S}} w_{\boldsymbol{S}}(e)\, \boldsymbol{\Delta}_e \in \mathbb{H}_{\mathsf{V}}.$$

It is rather irritating, but necessary, to keep track of which Laplacian we are operating on. The notation will reflect the choice.

Suppose that we wish to eliminate the vertex u from the Laplacian. Define the *star* induced by a vertex u is the Laplacian generated by the weighted edges in $\boldsymbol{S}$ that are incident on u. That is,

$$\textsc{star}(u, \boldsymbol{S}) = \sum_{e = uv \in \boldsymbol{S}} w_{\boldsymbol{S}}(e)\, \boldsymbol{\Delta}_e. \tag{6.2}$$

The sum takes place over all multiedges e in $\boldsymbol{S}$ that are incident on the vertex u. The *clique* induced by eliminating the vertex u from $\boldsymbol{S}$ is a weighted Laplacian

$$\textsc{clique}(u, \boldsymbol{S}) = \frac{1}{2 w_{\boldsymbol{S}}(u)} \sum_{e_1 = uv_1 \in \boldsymbol{S}} \sum_{e_2 = uv_2 \in \boldsymbol{S}} w_{\boldsymbol{S}}(e_1)\, w_{\boldsymbol{S}}(e_2)\, \boldsymbol{\Delta}_{v_1 v_2}. \tag{6.3}$$

Each sum takes place over all multiedges e in $\boldsymbol{S}$ that are incident on the vertex u.

The star is the Laplacian of a multigraph; the clique is also the Laplacian of a multigraph. By a direct calculation, the star and the clique satisfy the identity

$$\textsc{clique}(u, \boldsymbol{S}) - \textsc{star}(u, \boldsymbol{S}) = -\frac{1}{w_{\boldsymbol{S}}(u)} (\boldsymbol{S}\boldsymbol{\delta}_u)(\boldsymbol{S}\boldsymbol{\delta}_u)^*. \tag{6.4}$$

Therefore, the Schur complement $\boldsymbol{S}/u$ takes the form

$$\begin{aligned} \boldsymbol{S}/u &= \boldsymbol{S} - \frac{1}{w_{\boldsymbol{S}}(u)} (\boldsymbol{S}\boldsymbol{\delta}_u)(\boldsymbol{S}\boldsymbol{\delta}_u)^* \\ &= \big(\boldsymbol{S} - \textsc{star}(u, \boldsymbol{S})\big) + \textsc{clique}(u, \boldsymbol{S}). \end{aligned}$$

As before, the Schur complement $\boldsymbol{S}/u$ is the Laplacian of a multigraph. This multigraph has no edges incident on the vertex u. Moreover, if a vertex does not participate in $\boldsymbol{S}$, it does not participate in $\boldsymbol{S}/u$.

To repeat, we compute the Schur complement of $\boldsymbol{S}$ with respect to a vertex u by adding the clique induced by eliminating the vertex u from $\boldsymbol{S}$ and then removing the star induced by the vertex u from $\boldsymbol{S}$.

6.3.4 Cholesky decomposition of a Laplacian

We are now prepared to summarize the process of computing the (pivoted) Cholesky decomposition of the Laplacian $\boldsymbol{L}$ of the multigraph G.

Set $\boldsymbol{S}_0 = \boldsymbol{L}$. For each iteration $i = 1, 2, 3, \ldots, n$, select a vertex u_i. Extract the associated normalized column of the Laplacian:

$$\boldsymbol{c}_i = \frac{1}{\sqrt{(\boldsymbol{S}_{i-1})_{u_i u_i}}} \boldsymbol{S}_{i-1} \boldsymbol{\delta}_{u_i}.$$

Compute the Schur complement with respect to the vertex u_i:

$$\boldsymbol{S}_i = \boldsymbol{S}_{i-1} / u_i = \big(\boldsymbol{S}_{i-1} - \textsc{star}(u_i, \boldsymbol{S}_{i-1})\big) + \textsc{clique}(u_i, \boldsymbol{S}_{i-1}).$$

Let us emphasize that $\boldsymbol{S}_i$ remains a Laplacian matrix, but it has no multiedges incident on the vertices $\pi(1), \ldots, \pi(i)$. We have reduced the size of the problem, and the process continues.

After n steps, construct the morally lower-triangular matrix

$$\boldsymbol{C} = \begin{bmatrix} \boldsymbol{c}_1 & \ldots & \boldsymbol{c}_n \end{bmatrix} \in \mathbb{R}^{\mathsf{V} \times \mathsf{V}}.$$

Then the initial Laplacian admits the factorization

$$\boldsymbol{L} = \boldsymbol{C}\boldsymbol{C}^*.$$

Last, we record the permutation π defined by $\pi(i) = u_i$ for $i = 1, \ldots, n$. This permutation reflects the order in which the vertices were eliminated, and it is also determines the substitution order for solving the linear system $\boldsymbol{C}\boldsymbol{x} = \boldsymbol{f}$.

Let us remark that there is a standard approach to selecting vertices to eliminate from a graph Laplacian. At each step, we choose the remaining vertex that has the minimum degree.

6.3.5 An opportunity

Recall that the Cholesky decomposition is expensive because of the cost of computing the Schur complement. For general psd matrices, we compute the Schur complement by subtracting a rank-one matrix. It is not clear how to approximate this operation accurately.

For graph Laplacians, however, we expressed the Schur complement as the composition of two simple graph operations. Removing the star induced by a vertex is straightforward and inexpensive. The dominant cost arises from introducing the clique; this operation is quadratic in the number of edges incident on the vertex we eliminate.

Nevertheless, the clique is expressed as a weighted sum of many elementary Laplacians. As a consequence, we can try to approximate the clique by sampling. This is the core idea behind the SparseCholesky algorithm, which we detail in the Lecture 8.

The SparseCholesky is an iterative algorithm that constructs a sequence of random matrices. To analyze this kind of algorithm, we need more sophisticated matrix concentration tools. The next lecture turns to the subject of matrix martingales, which are the key to understanding the behavior of the algorithm.

7. Matrix Martingales

"Spielbank Wiesbaden," Wikimedia Commons

> Some of the text of this lecture is copied from my paper [Tro11a]. The treatment of corrector processes has not appeared before.

We plan to analyze a randomized, sequential algorithm that operates on matrices. For this purpose, we need to extend the theory of matrix concentration from independent sums to martingales. The purpose of this lecture is to present the main elements of this extension. In Lecture 8, we will require the full power of this approach.

7.1 Matrix-valued random processes

We begin with some basic definitions from the theory of random processes and their matrix-valued cousins.

7.1.1 Martingales

Let $(\Omega, \mathscr{F}, \mathbb{P})$ be a probability space, and let $\mathscr{F}_0 \subset \mathscr{F}_1 \subset \mathscr{F}_2 \subset \cdots \subset \mathscr{F}$ be a filtration of the master sigma algebra. We write $\mathbb{E}_k$ for the expectation conditioned on $\mathscr{F}_k$. That is, $\mathbb{E}_k$ averages over all random choices that take place after the instant k.

A *martingale* is a (real-valued) random process $\{Y_k : k = 0, 1, 2, \dots\}$ that is adapted to the filtration and that satisfies two properties:

$$\mathbb{E}_{k-1} Y_k = Y_{k-1} \quad \text{and} \quad \mathbb{E}\,|Y_k| < +\infty \quad \text{for } k = 1, 2, 3, \dots.$$

For simplicity, we require the initial value of a martingale to be null: $Y_0 = 0$. The *difference sequence* is the random process defined by

$$X_k = Y_k - Y_{k-1} \quad \text{for } k = 1, 2, 3, \dots.$$

Roughly, the present value of a martingale depends only on the past values, and the martingale has the *status quo* property: today, on average, is the same as yesterday.

A *supermartingale* is a (real-valued) random process $\{S_k : k \geq 0\}$ that is adapted to the filtration and that satisfies

$$\mathbb{E}_{k-1} S_k \leq S_{k-1}.$$

In other words, a supermartingale is a process with diminishing expectations.

7.1.2 Matrix martingales

Matrix martingales are defined in much the same manner as scalar martingales. Consider a random process $\{\boldsymbol{Y}_k : k = 0, 1, 2, \dots\}$ whose values are matrices of finite dimension. We say that the process is a *matrix martingale* when $\boldsymbol{Y}_0 = \mathbf{0}$ and

$$\mathbb{E}_{k-1} \boldsymbol{Y}_k = \boldsymbol{Y}_{k-1} \quad \text{and} \quad \mathbb{E} \|\boldsymbol{Y}_k\| < +\infty \quad \text{for } k = 1, 2, 3, \dots.$$

We write $\|\cdot\|$ for the *spectral norm* of a matrix, which returns its largest singular value. As before, we define the difference sequence $\{\boldsymbol{X}_k : k = 1, 2, 3, \dots\}$ via the relation

$$\boldsymbol{X}_k = \boldsymbol{Y}_k - \boldsymbol{Y}_{k-1} \quad \text{for } k = 1, 2, 3, \dots.$$

A matrix-valued random process is a martingale if and only if we obtain a scalar martingale when we track each fixed coordinate in time.

7.1.3 Adapted sequences

A sequence $\{\boldsymbol{X}_k\}$ of random matrices is *adapted* to the filtration when each $\boldsymbol{X}_k$ is measurable with respect to $\mathscr{F}_k$. That is, $\boldsymbol{X}_k$ is completely determined by random choices made up to and including instant k. We say that a sequence $\{\boldsymbol{V}_k\}$ of random matrices is *predictable* when each $\boldsymbol{V}_k$ is measurable with respect to $\mathscr{F}_{k-1}$. In particular, the sequence $\{\mathbb{E}_{k-1} \boldsymbol{X}_k\}$ of conditional expectations of an adapted sequence $\{\boldsymbol{X}_k\}$ is predictable. A *stopping time* is a random variable $K : \Omega \to \mathbb{N}_0 \cup \{\infty\}$ that satisfies

$$\{K \leq k\} \subset \mathscr{F}_k \quad \text{for } k = 0, 1, 2, \dots, \infty.$$

7.1.4 Stopped processes

Suppose that $\{S_k : k \geq 0\}$ is an adapted random process, and let K be a stopping time. The *stopped process* $\{S_{k \wedge K} : k \geq 0\}$ coincides with the original process up to the stopping time K, after which it remains constant.

Fact 7.1 (Stopped processes). Let $\{S_k\}$ be a (super)martingale, and let K be a stopping time. The stopped process $\{S_{k \wedge K}\}$ remains a (super)martingale. ∎

7.2 Tail bounds for matrix-valued processes

Now, let us develop a general methodology for establishing tail bounds for matrix-valued random processes. The basic technique can be traced at least as far as Freedman's work [Fre75] on scalar random processes. In the next section, we introduce the extra tools that are required to apply these results fruitfully in the matrix setting.

7.2.1 Corrector processes

We begin with the definition of a corrector process for a martingale. The corrector process is an auxiliary random process that provides an evolving bound on the growth of the martingale. This concept is rather abstract, but we will soon see how to make it more concrete.

Definition 7.2 (Corrector process). Let $g : [0, \infty] \to [0, \infty]$ be a function. Consider a martingale $\{\boldsymbol{Y}_k : k = 0, 1, 2, \dots\}$ and a predictable random process $\{\boldsymbol{W}_k : k = 0, 1, 2, \dots\}$ that consist of self-adjoint random matrices with dimension d. Define the real-valued random processes

$$S_k(\theta) = \operatorname{tr} \exp\left(\theta \boldsymbol{Y}_k - g(\theta)\, \boldsymbol{W}_k\right) \quad \text{for } \theta \geq 0. \tag{7.1}$$

We say that $\{g\, \boldsymbol{W}_k\}$ is a *corrector process* for the martingale $\{\boldsymbol{Y}_k\}$ if $S_k(\theta)$ is a positive supermartingale for all $\theta \geq 0$.

Since we are assuming that the martingale has a null initial value ($\boldsymbol{Y}_0 = \mathbf{0}$), it is natural to require that the corrector process also has null initial value ($\boldsymbol{W}_0 = \mathbf{0}$). In this case, the initial value of the supermartingale satisfies $S_0(\theta) = d$ for all $\theta \geq 0$. Furthermore, the supermartingale only takes positive values.

7.2.2 Lower bounds for the supermartingale

Next, we present a simple inequality that bounds the supermartingale S_k below when we have control on the eigenvalues of the two processes.

Lemma 7.3 Suppose that $\lambda_{\max}(\boldsymbol{Y}) \geq t$ and that $\lambda_{\max}(\boldsymbol{W}) \leq w$. For each $\theta > 0$,

$$\operatorname{tr} \exp(\theta \boldsymbol{Y} - g(\theta)\, \boldsymbol{W}) \geq \mathrm{e}^{\theta t - g(\theta)\, w}.$$

Proof. Recall that $g(\theta) \geq 0$. The bound results from a calculation:

$$\begin{aligned} \operatorname{tr} \mathrm{e}^{\theta \boldsymbol{Y} - g(\theta)\, \boldsymbol{W}} &\geq \operatorname{tr} \mathrm{e}^{\theta \boldsymbol{Y} - g(\theta)\, w \mathbf{I}} \\ &\geq \lambda_{\max}\left(\mathrm{e}^{\theta \boldsymbol{Y} - g(\theta)\, w \mathbf{I}}\right) = \mathrm{e}^{\theta \lambda_{\max}(\boldsymbol{Y}) - g(\theta)\, w} \geq \mathrm{e}^{\theta t - g(\theta)\, w}. \end{aligned}$$

The first inequality depends on the semidefinite relation $\boldsymbol{W} \preccurlyeq w\mathbf{I}$ and the monotonicity of the trace exponential with respect to the semidefinite order (Fact 1.8). The second inequality relies on the fact that the trace of a psd matrix is at least as large as its maximum eigenvalue. The third identity follows from the spectral mapping theorem and elementary properties of the maximum eigenvalue map. ∎

7.2.3 A tail bound for matrix martingales

Our key theorem provides a bound on the probability that the maximum eigenvalue of a matrix martingale ever exceeds a threshold.

Theorem 7.4 (Master tail bound for matrix martingales). Consider a matrix martingale $\{\boldsymbol{Y}_k\}$ consisting of self-adjoint matrices with dimension d. For a function $g : [0, \infty] \to [0, \infty]$, assume that $\{g\, \boldsymbol{W}_k\}$ is a corrector process for the martingale.

Then, for all $t, w \in \mathbb{R}$,

$$\mathbb{P}\{\exists k \geq 0 : \lambda_{\max}(\boldsymbol{Y}_k) \geq t \text{ and } \lambda_{\max}(\boldsymbol{W}_k) \leq w\} \leq d \cdot \inf_{\theta > 0} \mathrm{e}^{-\theta t + g(\theta) w}.$$

Proof. The overall proof strategy is the same as the stopping-time technique used by Freedman [Fre75]. Fix a positive parameter θ, which we will optimize later. Introduce the supermartingale $S_k = S_k(\theta)$, as in (7.1).

Define a stopping time K by finding the first time instant k when the maximum eigenvalue of the martingale reaches the level t even though the corrector process has maximum eigenvalue no larger than w. That is,

$$K := \inf\{k \geq 0 : \lambda_{\max}(\boldsymbol{Y}_k) \geq t \text{ and } \lambda_{\max}(\boldsymbol{W}_k) \leq w\}.$$

When the infimum is empty, the stopping time $K = \infty$. Consider a system of exceptional events:

$$\mathsf{E}_k := \{\lambda_{\max}(\boldsymbol{Y}_k) \geq t \text{ and } \lambda_{\max}(\boldsymbol{W}_k) \leq w\} \quad \text{for } k = 0, 1, 2, \dots.$$

Construct the event $\mathsf{E} := \bigcup_{k=0}^{\infty} \mathsf{E}_k$ that one or more of these exceptional situations takes place. The intuition behind this definition is that our control on the corrector process $\{\boldsymbol{W}_k\}$ prevents the martingale $\{\boldsymbol{Y}_k\}$ from exhibiting a large value. As a result, the event E is rather unlikely.

We are prepared to estimate the probability of the exceptional event. First, note that $K < \infty$ on the event E. Therefore, Lemma 7.3 provides a conditional lower bound for the supermartingale $\{S_k\}$ at the stopping time K:

$$S_K = \operatorname{tr} \exp\left(\theta \boldsymbol{Y}_K - g(\theta) \boldsymbol{W}_K\right) \geq \mathrm{e}^{\theta t - g(\theta) w} \quad \text{on the event } \mathsf{E}.$$

The stopped process $\{S_{k \wedge K}\}$ is also a positive supermartingale with initial value d, so

$$d \geq \liminf_{k \to \infty} \mathbb{E}[S_{k \wedge K}] \geq \liminf_{k \to \infty} \mathbb{E}[S_{k \wedge K} \mathbb{1}_{\mathsf{E}}] \geq \mathbb{E}[\liminf_{k \to \infty} S_{k \wedge K} \mathbb{1}_{\mathsf{E}}] = \mathbb{E}[S_K \mathbb{1}_{\mathsf{E}}].$$

The indicator function decreases the expectation because the stopped process is positive. Fatou's lemma justifies the third inequality, and we have identified the limit using the fact that $K < \infty$ on the event E. It follows that

$$d \geq \mathbb{E}[S_K \mathbb{1}_{\mathsf{E}}] \geq (\mathbb{P}\mathsf{E}) \cdot \inf_{\mathsf{E}} S_K \geq (\mathbb{P}\mathsf{E}) \cdot \mathrm{e}^{\theta t - g(\theta) w}.$$

Rearrange the relation to obtain

$$\mathbb{P}\mathsf{E} \leq d \cdot \mathrm{e}^{-\theta t + g(\theta) w}.$$

Minimize the right-hand side with respect to θ to complete the main part of the argument. ∎

7.3 Building a corrector process

To convert Theorem 7.4 into a useful tool, we need a mechanism for constructing a corrector process. Fortunately, as in the case of independent sums of random matrices, Lieb's theorem comes to our rescue. We will see that we can construct a corrector process using matrix cgfs.

7.3.1 Correctors

Let us specialize the notion of a corrector to a single matrix. This will be the building block for constructing a corrector process.

Definition 7.5 (Corrector). Let $g : [0, \infty] \to [0, \infty]$ be a function. Consider a random self-adjoint matrix $\boldsymbol{X}$ and a fixed matrix $\boldsymbol{V}$, each with dimension d. We say that $g\,\boldsymbol{V}$ is a *corrector* for $\boldsymbol{X}$ when

$$\mathbb{E}\operatorname{tr}\exp(\boldsymbol{M} + \theta\boldsymbol{X} - g(\theta)\,\boldsymbol{V}) \le \operatorname{tr}\exp(\boldsymbol{M}) \quad \text{for } \theta > 0.$$

This bound must hold for every fixed matrix $\boldsymbol{M} \in \mathbb{H}_d$.

7.3.2 Lieb's theorem and Tropp's corollary

Our main tool for producing explicit correctors is Lieb's theorem [Lie73, Thm. 6]. We refer to [Tro15, Chap. 8] for a digestible proof of this result.

Theorem 7.6 (Lieb, 1973). Fix a self-adjoint matrix $\boldsymbol{H}$. The function

$$\boldsymbol{A} \longmapsto \operatorname{tr}\exp(\boldsymbol{H} + \log \boldsymbol{A})$$

is concave on the pd cone.

Lieb's theorem tells us that we can construct a corrector from a cumulant generating function. This simple but powerful observation first appeared in [Tro11a].

Corollary 7.7 (Tropp, 2010). Let $\boldsymbol{M}$ be a fixed self-adjoint matrix, and let $\boldsymbol{X}$ be a random self-adjoint matrix of the same dimension. For any $\theta \in \mathbb{R}$,

$$\mathbb{E}\operatorname{tr}\exp\left(\boldsymbol{M} + \theta\boldsymbol{X} - \log \mathbb{E}\,\mathrm{e}^{\theta\boldsymbol{X}}\right) \le \mathbb{E}\operatorname{tr}\exp(\boldsymbol{M}).$$

Proof. Define the random matrix $\boldsymbol{Y} = \mathrm{e}^{\theta\boldsymbol{X}}$, and calculate that

$$\begin{aligned}\mathbb{E}\operatorname{tr}\exp\left(\boldsymbol{M} + \theta\boldsymbol{X} - \log \mathbb{E}\,\mathrm{e}^{\theta\boldsymbol{X}}\right) &= \mathbb{E}\operatorname{tr}\exp\left(\boldsymbol{M} + \log(\boldsymbol{Y}) - \log(\mathbb{E}\,\boldsymbol{Y})\right)\\ &\le \operatorname{tr}\exp\left(\boldsymbol{M} + \log(\mathbb{E}\,\boldsymbol{Y}) - \log(\mathbb{E}\,\boldsymbol{Y})\right)\\ &= \operatorname{tr}\exp(\boldsymbol{M}).\end{aligned}$$

The first identity follows because the logarithm of the pd matrix $\boldsymbol{Y}$ can be defined as the functional inverse of the matrix exponential. Theorem 7.6, with the fixed matrix $\boldsymbol{H} = \boldsymbol{M} - \log(\mathbb{E}\,\boldsymbol{Y})$, establishes that the trace function is concave in $\boldsymbol{Y}$. Invoke Jensen's inequality to draw the expectation inside the logarithm. ∎

7.3.3 Example: The Bernstein corrector

Corollary 7.7 and Lemma 1.10 allow us to derive a corrector for a bounded, centered random matrix.

Proposition 7.8 (Bernstein corrector). Let $\boldsymbol{X}$ be a random matrix that satisfies $\mathbb{E}\,\boldsymbol{X} = \boldsymbol{0}$ and $\|\boldsymbol{X}\| \le 1$. Then the matrix $g(\theta)(\mathbb{E}\,\boldsymbol{X}^2)$ is a corrector for $\boldsymbol{X}$, where $g(\theta) = (\theta^2/2)/(1 - |\theta|/3)$.

Proof. We may calculate that

$$\mathbb{E}\operatorname{tr}\exp\left(\boldsymbol{M}+\theta\boldsymbol{X}-g(\theta)(\mathbb{E}\,\boldsymbol{X}^2)\right)\le\mathbb{E}\operatorname{tr}\exp\left(\boldsymbol{M}+\theta\boldsymbol{X}-\log\mathbb{E}\,\mathrm{e}^{\theta\boldsymbol{X}}\right)\le\operatorname{tr}\exp(\boldsymbol{M}).$$

The first inequality follows from the Bernstein cgf bound, Lemma 1.10, because the trace exponential is monotone with respect to the semidefinite order (Fact 1.8). The second inequality is Corollary 7.7. ∎

7.3.4 Example: The Chernoff corrector

Corollary 7.7 and Lemma 1.12 allow us to derive a corrector for a bounded, psd random matrix.

Proposition 7.9 (Chernoff corrector). Let $\boldsymbol{X}$ be a random matrix that satisfies the bounds $\mathbf{0}\preccurlyeq\boldsymbol{X}\preccurlyeq\mathbf{I}$. Then the matrix $g(\theta)(\mathbb{E}\,\boldsymbol{X})$ is a corrector for $\boldsymbol{X}$, where $g(\theta)=\mathrm{e}^{\theta}-1$.

We omit the repetitive proof.

7.3.5 From correctors to corrector processes

There is a straightforward connection between the corrector of a single random and the corrector process of a martingale.

Proposition 7.10 (Corrector processes). Fix a function $g:[0,\infty]\to[0,\infty]$. Let $\{\boldsymbol{Y}_k\}$ be a self-adjoint matrix martingale with difference sequence $\{\boldsymbol{X}_k\}$. Let $\{\boldsymbol{V}_k\}$ be a predictable sequence of self-adjoint matrices. For each k, suppose that $g\,\boldsymbol{V}_k$ is a corrector for $\boldsymbol{X}_k$, conditional on $\mathscr{F}_{k-1}$. Then the predictable process

$$\boldsymbol{W}_k=\sum\nolimits_{i=1}^{k}\boldsymbol{V}_i$$

generates a corrector $\{g\,\boldsymbol{W}_k\}$ for the martingale $\{\boldsymbol{Y}_k\}$.

Proof. As above, define

$$S_k(\theta)=\operatorname{tr}\exp(\theta\boldsymbol{Y}_k-g(\theta)\,\boldsymbol{W}_k).$$

To prove that the process is a supermartingale, we follow a short chain of inequalities. Split off the last term from $\boldsymbol{Y}_k$ and $\boldsymbol{W}_k$ to see that

$$\begin{aligned}\mathbb{E}_{k-1}\,S_k(\theta)&=\mathbb{E}_{k-1}\operatorname{tr}\exp\left(\theta\boldsymbol{Y}_{k-1}-g(\theta)\,\boldsymbol{W}_{k-1}+\theta\boldsymbol{X}_k-g(\theta)\,\boldsymbol{V}_k\right)\\&\le\operatorname{tr}\exp\left(\theta\boldsymbol{Y}_{k-1}-g(\theta)\,\boldsymbol{W}_{k-1}\right)=S_{k-1}(\theta).\end{aligned}$$

This inequality follows immediately from the assumption that $g\,\boldsymbol{V}_k$ is a corrector for $\boldsymbol{X}_k$, conditional on $\mathscr{F}_{k-1}$. We can apply this hypothesis because $\boldsymbol{Y}_{k-1}$ and $\boldsymbol{W}_{k-1}$ both are measurable with respect to $\mathscr{F}_{k-1}$. ∎

7.3.6 Correctors tensorize

Let us continue with some general methods for constructing correctors of more complicated matrices. First, correctors tensorize over independent random matrices.

Proposition 7.11 (Correctors tensorize). Let $g : [0, \infty] \to [0, \infty]$. Consider an independent family $\{\boldsymbol{X}_k : k = 1, \dots, n\}$ of self-adjoint random matrices, and a nonrandom family $\{\boldsymbol{V}_k : k = 1, \dots, n\}$ of self-adjoint matrices. Suppose that $g\,\boldsymbol{V}_k$ is a corrector for $\boldsymbol{X}_k$ for each k. Then $g\,\sum_k \boldsymbol{V}_k$ is a corrector for $\sum_k \boldsymbol{X}_k$.

Proof. This result follows by iteration of Definition 7.5. Let $\boldsymbol{M}$ be a fixed matrix.

$$\begin{aligned}
&\mathbb{E}\,\mathbb{E}_n \operatorname{tr}\exp\Big(\boldsymbol{M} + \theta \sum\nolimits_{k=1}^{n} \boldsymbol{X}_k - g(\theta) \sum\nolimits_{k=1}^{n} \boldsymbol{V}_k\Big) \\
&\qquad \le \mathbb{E}\,\mathbb{E}_{n-1} \operatorname{tr}\exp\Big(\boldsymbol{M} + \theta \sum\nolimits_{k=1}^{n-1} \boldsymbol{X}_k - g(\theta) \sum\nolimits_{k=1}^{n-1} \boldsymbol{V}_k\Big) \\
&\qquad \le \cdots \le \operatorname{tr}\exp(\boldsymbol{M}).
\end{aligned}$$

This is what we needed to show. ∎

7.3.7 The composition rule

Next, let us present a composition rule that allows us to derive a corrector for a random matrix that is constructed in multiple steps.

Proposition 7.12 (Composition rule). Consider sigma fields $\mathscr{F}_0 \subset \mathscr{F}_1 \subset \mathscr{F}_2$. Let $\boldsymbol{X}$ be a random matrix that is measurable with respect to $\mathscr{F}_2$. For $\theta \ge 0$, suppose that

$$\begin{aligned}
\mathbb{E}\left[\operatorname{tr}\exp\left(\boldsymbol{M}_1 + \theta\boldsymbol{X} - g(\theta)\,\boldsymbol{V}_1\right) \mid \mathscr{F}_1\right] &\le \operatorname{tr}\exp\left(\boldsymbol{M}_1\right) \\
\mathbb{E}\operatorname{tr}\exp\left(\boldsymbol{M}_0 + \theta\boldsymbol{V}_1 - h(\theta)\,\boldsymbol{V}_0\right) &\le \operatorname{tr}\exp\left(\boldsymbol{M}_0\right).
\end{aligned}$$

In this expression, $\boldsymbol{V}_1$ and $\boldsymbol{M}_1$ are measurable with respect to $\mathscr{F}_1$, while $\boldsymbol{V}_0$ and $\boldsymbol{M}_0$ are measurable with respect to $\mathscr{F}_0$. Then $(h \circ g)\,\boldsymbol{V}_0$ is a corrector for $\boldsymbol{X}$.

Proof. Let $\boldsymbol{M}$ be measurable with respect to $\mathscr{F}_0$. Calculate that

$$\begin{aligned}
&\mathbb{E}\operatorname{tr}\exp\big(\boldsymbol{M} + \theta\boldsymbol{X} - (h \circ g)(\theta)\,\boldsymbol{V}_0\big) \\
&\qquad = \mathbb{E}\,\mathbb{E}\left[\operatorname{tr}\exp\left(\boldsymbol{M} + \theta\boldsymbol{X} - g(\theta)\,\boldsymbol{V}_1 + g(\theta)\,\boldsymbol{V}_1 - (h \circ g)(\theta)\,\boldsymbol{V}_0\right) \,\middle|\, \mathscr{F}_1\right] \\
&\qquad \le \mathbb{E}\operatorname{tr}\exp\left(\boldsymbol{M} + g(\theta)\,\boldsymbol{V}_1 - h(g(\theta))\,\boldsymbol{V}_0\right) \\
&\qquad \le \operatorname{tr}\exp(\boldsymbol{M}).
\end{aligned}$$

This is the definition of a corrector. ∎

7.4 Example: The matrix Freedman inequality

As an example, let us use Theorem 7.4 to prove the matrix version of a classic martingale inequality due to Freedman.

Theorem 7.13 (Matrix Freedman). Consider a matrix martingale $\{\boldsymbol{Y}_k\}$ consisting of self-adjoint matrices with dimension d. Assume that the difference sequence $\{\boldsymbol{X}_k\}$ satisfies

$$\|\boldsymbol{X}_k\| \le B \quad \text{for } k = 1, 2, 3, \dots.$$

Define the cumulative predictable quadratic variation process:

$$\boldsymbol{W}_0 = \mathbf{0} \quad \text{and} \quad \boldsymbol{W}_k = \sum\nolimits_{i=1}^{k} \mathbb{E}_{i-1} \boldsymbol{X}_i^2 \quad \text{for } k = 1, 2, 3, \dots.$$

Then, for all $t \geq 0$ and $\sigma^2 \geq 0$,

$$\mathbb{P}\left\{\exists k \geq 0 : \lambda_{\max}(\boldsymbol{Y}_k) \geq t \ \text{ and } \ \lambda_{\max}(\boldsymbol{W}_k) \leq \sigma^2\right\} \leq d \cdot \exp\left(\frac{-t^2/2}{\sigma^2 + Bt/3}\right).$$

Proof. We assume that $B = 1$; the general result follows by re-scaling since $\boldsymbol{Y}_k$ is 1-homogeneous and $\boldsymbol{W}_k$ is 2-homogeneous.

Invoke Proposition 7.8 conditionally to obtain a corrector process for $\{\boldsymbol{X}_k\}$. Indeed, we can choose

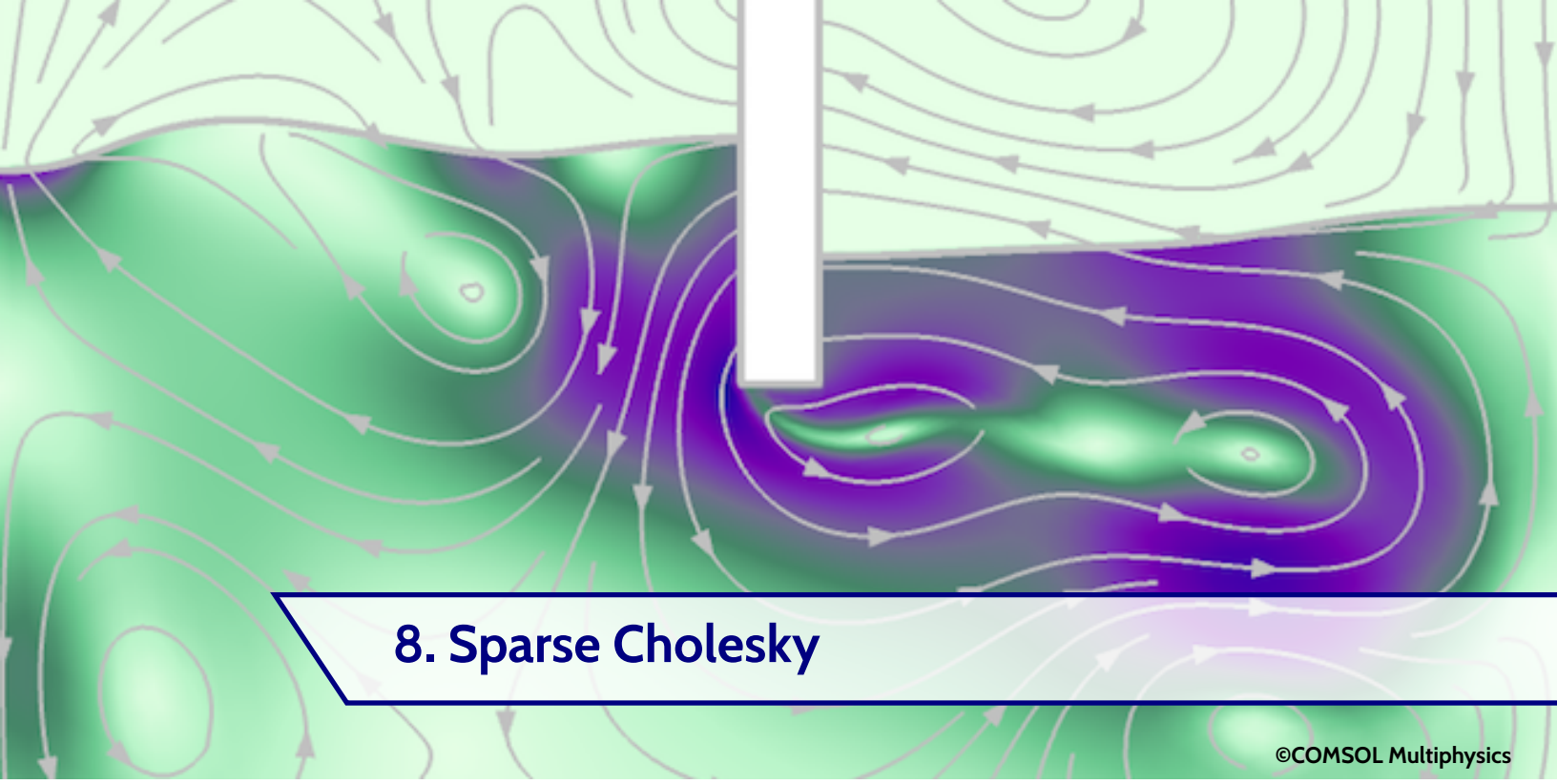
$$\boldsymbol{V}_k = \mathbb{E}_{k-1} \boldsymbol{X}_k^2 \quad \text{and} \quad g(\theta) = \frac{\theta^2/2}{1 - |\theta|/3}.$$

Theorem 7.4 now implies that

$$\mathbb{P}\left\{\exists k \geq 0 : \lambda_{\max}(\boldsymbol{Y}_k) \geq t \ \text{ and } \ \lambda_{\max}(\boldsymbol{W}_k) \leq \sigma^2\right\} \leq d \cdot \inf_{\theta > 0} \mathrm{e}^{-\theta t + g(\theta)\sigma^2}.$$

Make the inspired choice $\theta = t/(\sigma^2 + t/3)$ to complete the argument. ∎

Exercise 7.1 Extend the matrix Freedman inequality to a martingale sequence consisting of rectangular matrices.



8. Sparse Cholesky

This lecture is adapted from Rasmus Kyng's thesis [Kyn17]. The application of matrix martingales has been streamlined by using the notion of a corrector process.

In this lecture, we introduce a practical algorithm for solving Laplacian linear systems in near-linear time. The algorithm is remarkable in its simplicity, but the analysis relies on many of the sophisticated ideas that we have encountered in the previous lectures.

This approach, called the SparseCholesky algorithm, was developed by Rasmus Kyng and Sushant Sachdeva [KS16]. It was further refined in Kyng's dissertation [Kyn17]. It is closely related to an earlier algorithm for connection Laplacians, developed by Dan Spielman's group [Kyn+16]. Altogether, these methods hold real promise for solving large graph Laplacian systems in practice.

8.1 Approximate solutions of Laplacian systems

We begin with a high-level approach for computing an approximation solution of a Laplacian system via preconditioning.

8.1.1 Approximate solutions

Let $\boldsymbol{L}$ be the Laplacian matrix of a connected multigraph. Suppose that we wish to find the unique solution $\boldsymbol{x}_\star$ to the linear system

$$\boldsymbol{L}\boldsymbol{x} = \boldsymbol{f} \quad \text{where} \quad \mathbf{1}^* \boldsymbol{f} = \mathbf{0} \quad \text{and} \quad \mathbf{1}^* \boldsymbol{x} = \mathbf{0}.$$

For a parameter $\varepsilon > 0$, we can relax our requirement by asking for an approximate solution $\boldsymbol{x}_\varepsilon$ that satisfies the relative error bound

$$\|\boldsymbol{x}_\varepsilon - \boldsymbol{x}_\star\|_{\boldsymbol{L}} \le \varepsilon \cdot \|\boldsymbol{x}_\star\|_{\boldsymbol{L}}$$

Here, $\|\cdot\|_{\boldsymbol{L}}$ is the norm associated with the quadratic form (i.e., the Dirichlet form) determined by $\boldsymbol{L}$. That is, $\|\boldsymbol{x}\|_{\boldsymbol{L}} = (\boldsymbol{x}^* \boldsymbol{L} \boldsymbol{x})^{1/2}$.

8.1.2 Approximate Cholesky decomposition

Suppose that we are able to construct a sparse, approximate Cholesky decomposition of the Laplacian matrix:

$$0.5\,\boldsymbol{L} \preccurlyeq \boldsymbol{C}\boldsymbol{C}^* \preccurlyeq 1.5\,\boldsymbol{L} \quad \text{where} \quad \mathsf{nnz}(\boldsymbol{C}) = O(m \log n). \tag{8.1}$$

The symbol $\preccurlyeq$ refers to the semidefinite order. The matrix $\boldsymbol{C}$ is morally lower-triangular; in other words, there is a permutation of coordinates that brings the matrix into lower-triangular form. The function nnz returns the number of nonzero entries in a matrix.

8.1.3 Preconditioning

Given the sparse, approximate factor $\boldsymbol{C}$, we can precondition the linear system (6.1):

$$(\boldsymbol{C}^\dagger \boldsymbol{L} \boldsymbol{C}^{*\dagger})(\boldsymbol{C}^* \boldsymbol{x}) = (\boldsymbol{C}^\dagger \boldsymbol{f}).$$

Owing to (8.1), the preconditioned linear system has condition number $\kappa \le 3$. Of course, in practice, we treat the matrix as an operator acting on vectors. Each time we apply the operator, we use forward and back substitution to invoke $\boldsymbol{C}^\dagger$ and $\boldsymbol{C}^{*\dagger}$. The total cost of each application of the matrix is $\Theta(m \log n)$ arithmetic operations, because the substitution method for solving a morally triangular system exploits sparsity.

We can solve the preconditioned system using the conjugate gradient algorithm. If the initial iterate $\boldsymbol{x}_0 = \boldsymbol{0}$, then, after j iterations, we attain the error bound

$$\|\boldsymbol{x}_j - \boldsymbol{x}_\star\|_{\boldsymbol{L}} \le 2 \left[\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right]^j \|\boldsymbol{x}_\star\|_{\boldsymbol{L}} \le 3^{1-j} \|\boldsymbol{x}_\star\|_{\boldsymbol{L}}.$$

In particular, we can achieve a relative error of ε in the Dirichlet energy norm after $O(\log(1/\varepsilon))$ iterations.

8.1.4 Summary

In summary, once we have constructed an approximate Cholesky decomposition that satisfies (8.1), we can solve the linear system to relative error ε using $O(m \log(n) \log(1/\varepsilon))$ arithmetic operations. This computation takes place in time that is nearly linear in the number of degrees of freedom in the graph.

Easy! We just have to achieve (8.1). In the rest of this lecture, we will explain how to perform this feat.

8.2 Overview of the algorithm

Let us begin with an overview of the SparseCholesky algorithm for computing a sparse, approximate Cholesky decomposition that satisfies (8.1). We will fill in the details and perform the analysis over the balance of the lecture.

8.2.1 Setup

Fix the ground set $\mathsf{V} = \{1, \dots, n\}$ of vertices. Let G be a connected multigraph on V composed of m weighted multiedges. As usual, we will interact with the multigraph G via its Laplacian matrix $\boldsymbol{L}$. The Laplacian will be represented as a sum over multiedges:

$$\boldsymbol{L} = \sum\nolimits_{e \in \boldsymbol{L}} w_{\boldsymbol{L}}(e)\, \boldsymbol{\Delta}_e \in \mathbb{H}_{\mathsf{V}}.$$

The indexing for the sum and the notation for the weight function are intended to be mnemonic, if not overly formal.

8.2.2 The SparseCholesky procedure

The SparseCholesky algorithm is based on the same template as the ordinary Cholesky decomposition, but it judiciously injects randomness to minimize the computational burden. The basic idea is to randomly sample the cliques that arise as we eliminate vertices from the multigraph:

Summary of SparseCholesky:

For each iteration $i = 1, \dots, n$:

1. Select a random vertex u_i to eliminate.
2. Add a random approximation of the clique induced by u_i.
3. Remove the star induced by u_i.

We continue with a more detailed presentation of the steps in the procedure.

Preprocessing

Before we begin, we split each multiedge into a fixed number of pieces to reduce the leverage of each multiedge below a threshold.

As the algorithm constructs new multiedges, we will ensure that the leverages never increase beyond the initial threshold. This property helps control the variance of the random clique approximations.

Initialization

Let $\boldsymbol{S}_0 = \boldsymbol{L}$. Let $\mathsf{F}_0 = \mathsf{V}$ of vertices that have not been eliminated.

We will maintain the invariant that the iterate $\boldsymbol{S}_i$ is supported on the vertices listed in F_i. The number of vertices remaining at each step will satisfy $|\mathsf{F}_i| = n - i$.

Selecting a vertex to eliminate

At each iteration $i = 1, 2, 3, \dots, n$, select a vertex u_i uniformly at random from F_{i-1}. Update $\mathsf{F}_i = \mathsf{F}_{i-1} \setminus \{u_i\}$.

Selecting a random vertex u_i renders it unlikely that there are many multiedges incident on u_i. Furthermore, it is unlikely that the clique induced by u_i involves multiedges whose total leverage is large. These facts are critical for controlling the runtime of the algorithm and ensuring that it produces an accurate approximation.

Collecting information

Extract data from the current iterate $\boldsymbol{S}_{i-1}$:

$$\boldsymbol{c}_i = \frac{1}{\sqrt{(\boldsymbol{S}_{i-1})_{u_i u_i}}} \boldsymbol{S}_{i-1} \boldsymbol{\delta}_{u_i}.$$

If $(\boldsymbol{S}_{i-1})_{u_i u_i} = 0$, then we set $\boldsymbol{c}_i = \mathbf{0}$. Since the matrix $\boldsymbol{S}_{i-1}$ is supported on the coordinates listed in F_{i-1}, the support of $\boldsymbol{c}_i$ is also contained in F_{i-1}.

Sampling the clique

To proceed, we will approximate the Schur complement $\boldsymbol{S}_{i-1}/u_i$. To do so, we first construct a random sparse Laplacian matrix $\boldsymbol{K}_i$ that approximates the clique induced by eliminating u_i. We will explain how to perform this approximation later, in Section 8.4. The basic requirement on that $\boldsymbol{K}_i$ is that

$$\mathbb{E}_{i-1}[\boldsymbol{K}_i \mid u_i] = \textsc{clique}(u_i, \boldsymbol{S}_{i-1}).$$

The expectation $\mathbb{E}_{i-1}$ conditions on all of the randomness in the first $i-1$ iterations. We also condition separately on the random vertex u_i drawn at step i.

The number of multiedges in the clique approximation $\boldsymbol{K}_i$ will not exceed the total number of multiedges incident on the vertex u_i, so the number of multiedges remaining in the multigraph does not increase as the iteration advances. This property also ensures that the cost of computing the clique approximation is under control.

Moreover, we will ensure that the clique approximation $\boldsymbol{K}_i$ has no multiedges incident on $u_1, \dots, u_i$. That is, $\boldsymbol{K}_i$ is supported on the coordinates listed in F_i.

Approximating the Schur complement

Now, form the approximate Schur complement:

$$\boldsymbol{S}_i = \big(\boldsymbol{S}_{i-1} - \textsc{star}(u_i, \boldsymbol{S}_{i-1})\big) + \boldsymbol{K}_i. \tag{8.2}$$

In the last step, we set $\boldsymbol{S}_n = \mathbf{0}$.

For reference, the star (6.2) and clique (6.3) induced by a vertex were defined before. This construction ensures that $\boldsymbol{S}_i$ is supported on the coordinates listed in F_i. Therefore, we continue to reduce the size of the problem.

Forming the decomposition

As usual, we conclude by compiling the matrix

$$\boldsymbol{C} = \begin{bmatrix} \boldsymbol{c}_1 & \dots & \boldsymbol{c}_n \end{bmatrix} \in \mathbb{R}^{\mathsf{V} \times \mathsf{V}}.$$

By construction of the vectors $\boldsymbol{c}_i$, the matrix $\boldsymbol{C}$ is morally lower-triangular. The elimination order is associated with the permutation π defined by $\pi(i) = u_i$ for $i = 1, \dots, n$.

8.2.3 Laplacian approximations

How do we make sense of this approach? Note that the SparseCholesky iteration induces a sequence of approximations to the Laplacian matrix:

$$\boldsymbol{L}_i = \boldsymbol{S}_i + \sum\nolimits_{k=1}^{i} \boldsymbol{c}_k \boldsymbol{c}_k^* \quad \text{for } i = 0, 1, 2, \dots, n.$$

In particular, the initial value of the sequence is the original Laplacian, while the final value is our approximate Cholesky decomposition:

$$\boldsymbol{L}_0 = \boldsymbol{L} \quad \text{and} \quad \boldsymbol{L}_n = \sum\nolimits_{i=1}^{n} \boldsymbol{c}_i \boldsymbol{c}_i^* = \boldsymbol{C}\boldsymbol{C}^*.$$

The difference sequence of the random process $\{\boldsymbol{L}_i\}$ satisfies

$$\begin{aligned} \boldsymbol{L}_i - \boldsymbol{L}_{i-1} &= \boldsymbol{S}_i - \boldsymbol{S}_{i-1} + \boldsymbol{c}_i \boldsymbol{c}_i^* \\ &= \boldsymbol{K}_i - \textsc{star}(u_i, \boldsymbol{S}_{i-1}) + \boldsymbol{c}_i \boldsymbol{c}_i^* \\ &= \boldsymbol{K}_i - \textsc{clique}(u_i, \boldsymbol{S}_{i-1}) \\ &= \boldsymbol{K}_i - \mathbb{E}[\boldsymbol{K}_i \mid u_i]. \end{aligned}$$

The second relation follows from the definition (8.2) of the approximate Schur complement $\boldsymbol{S}_i$. We have used an identity from the last lecture:

$$\textsc{clique}(u_i, \boldsymbol{S}_{i-1}) = \textsc{star}(u_i, \boldsymbol{S}_{i-1}) - \boldsymbol{c}_i \boldsymbol{c}_i^*. \tag{8.3}$$

It is now evident that each increment is conditionally zero mean:

$$\mathbb{E}_{i-1}[\boldsymbol{L}_i - \boldsymbol{L}_{i-1}] = \mathbb{E}_{i-1}\, \mathbb{E}_{i-1}[\boldsymbol{L}_i - \boldsymbol{L}_{i-1} \mid u_i] = \boldsymbol{0}.$$

In particular,

$$\mathbb{E}\, \boldsymbol{L}_i = \boldsymbol{L} \quad \text{for each } i = 1, 2, 3, \dots, n.$$

We discover that $\{\boldsymbol{L}_i - \boldsymbol{L}_0\}$ is a matrix martingale with null initial value. The final value of this martingale is the error in the approximate Cholesky decomposition:

$$\boldsymbol{C}\boldsymbol{C}^* - \boldsymbol{L} = \boldsymbol{L}_n - \boldsymbol{L}_0 = \sum\nolimits_{i=1}^{n} (\boldsymbol{L}_i - \boldsymbol{L}_{i-1}).$$

Therefore, we can use the theory of matrix martingales to understand the behavior of the algorithm.

8.3 Preliminaries for the analysis

Let us begin the argument with some preliminary notation and simplifications.

8.3.1 The normalizing map

We define the normalizing map $\boldsymbol{\Phi}$ associated with the Laplacian $\boldsymbol{L}$ of the initial multigraph G:

$$\boldsymbol{\Phi}(\boldsymbol{M}) = (\boldsymbol{L}^\dagger)^{1/2} \boldsymbol{M} (\boldsymbol{L}^\dagger)^{1/2} \quad \text{for } \boldsymbol{M} \in \mathbb{H}_V.$$

This map has two properties that will play a role in the argument. Since G is connected, $\boldsymbol{\Phi}(\boldsymbol{L}) = \boldsymbol{P}$, where $\boldsymbol{P}$ is the orthogonal projector onto $\operatorname{lin}\{\mathbf{1}\}^{\perp}$. Second, $\boldsymbol{\Phi}$ is a *positive map*. That is,

$$\boldsymbol{M} \succcurlyeq \mathbf{0} \quad \text{implies} \quad \boldsymbol{\Phi}(\boldsymbol{M}) \succcurlyeq \mathbf{0}.$$

Let us emphasize that $\boldsymbol{\Phi}$ is always constructed from the Laplacian $\boldsymbol{L}$ of the initial multigraph.

8.3.2 The approximation requirement

Recall that the random process $\{\boldsymbol{L}_i\}$ has the terminal value $\boldsymbol{L}_n = \boldsymbol{C}\boldsymbol{C}^*$. We can express the approximation requirement (8.1) as

$$-0.5\,\boldsymbol{L} \preccurlyeq \boldsymbol{L}_n - \boldsymbol{L} \preccurlyeq +0.5\,\boldsymbol{L}.$$

Since $\boldsymbol{L}_n$ is a Laplacian, its range is contained in the range of the Laplacian $\boldsymbol{L}$. Therefore, we can apply the normalizing map to obtain an equivalent condition:

$$-0.5\,\boldsymbol{P} \preccurlyeq \boldsymbol{\Phi}(\boldsymbol{L}_n - \boldsymbol{L}) \preccurlyeq +0.5\,\boldsymbol{P}.$$

Using the relation $\boldsymbol{L}_0 = \boldsymbol{L}$ and taking care with the ranges of the matrices that appear, we can write the latter expression as a pair of eigenvalue bounds:

$$\begin{aligned}\lambda_{\max}(\boldsymbol{\Phi}(\boldsymbol{L}_n - \boldsymbol{L}_0)) &\le +0.5;\\ \lambda_{\min}(\boldsymbol{\Phi}(\boldsymbol{L}_n - \boldsymbol{L}_0)) &\ge -0.5.\end{aligned}$$

In other words, we must control the discrepancy between the terminal value $\boldsymbol{L}_n$ and the initial value $\boldsymbol{L}_0$ of the random process. Matrix martingale inequalities are tailor-made for this purpose.

To see how this will work, we decompose the martingale into its difference sequence:

$$\begin{aligned}\boldsymbol{\Phi}(\boldsymbol{L}_n - \boldsymbol{L}_0) &= \sum\nolimits_{i=0}^{n} \boldsymbol{\Phi}(\boldsymbol{L}_i - \boldsymbol{L}_{i-1})\\ &= \sum\nolimits_{i=0}^{n} \boldsymbol{\Phi}(\boldsymbol{K}_i - \textsc{clique}(u_i, \boldsymbol{S}_{i-1}))\\ &= \sum\nolimits_{i=0}^{n} \boldsymbol{\Phi}(\boldsymbol{K}_i - \mathbb{E}_{i-1}[\boldsymbol{K}_i \mid u_i]).\end{aligned}$$

The next step is to construct and analyze the randomized clique estimators $\boldsymbol{K}_i$. As a result, we will obtain a corrector process for the martingale $\{\boldsymbol{L}_i\}$, which will lead to the required tail bounds.

8.3.3 Splitting the edges

Recall that many matrix concentration bounds, such as the matrix Bernstein and matrix Freedman inequalities, require some type of uniform control over the random contributions. To obtain this control, we will preprocess the multigraph by splitting each multiedge into pieces.

Let $R \ge 1$ be a parameter that we will fix later. (To be concrete, we will set $R = \Theta(\log n)$.) We construct a new multigraph, with Laplacian $\boldsymbol{L}'$, by splitting each

edge in the Laplacian $\boldsymbol{L}$ into R equal pieces. This action has the effect of multiplying each leverage by a factor of $1/R$.

More precisely, we iterate over each multiedge $e = uv$ in $\boldsymbol{L}$; its weight is denoted as $w_{\boldsymbol{L}}(e)$. We augment the new Laplacian $\boldsymbol{L}'$ with R edges:

$$e_j = uv \quad \text{with} \quad w_{\boldsymbol{L}'}(e_j) = \frac{1}{R}\, w_{\boldsymbol{L}}(e) \quad \text{for each } j = 1, 2, 3, \dots, R.$$

As matrices, the Laplacians are equal: $\boldsymbol{L}' = \boldsymbol{L}$. Regarded as multigraphs, $\boldsymbol{L}'$ now has $M = Rm$ multiedges, whereas $\boldsymbol{L}$ only has m multiedges.

The leverage of each multiedge with respect to the new multigraph satisfies

$$w_{\boldsymbol{L}'}(e_j)\, \varrho_{\boldsymbol{L}'}(u, v) = \frac{1}{R}\, w_{\boldsymbol{L}}(e)\, \varrho_{\boldsymbol{L}}(u, v) \le \frac{1}{R}.$$

Indeed, since the Laplacians are equal, the effective resistance of each pair of vertices is the same in both graphs. The last identity holds because of Proposition 5.6. Every multiedge that we construct during the algorithm will satisfy this same bound.

To avoid an extra notational burden, we will simply assume that the input Laplacian $\boldsymbol{L}$ consists of $M = mR$ multiedges, each with leverage score bounded by $1/R$. Effective resistances ϱ will always be computed with respect to this Laplacian $\boldsymbol{L}$.

8.4 Sampling from a clique

The main challenge in the SparseCholesky algorithm is to avoid the cost of constructing the full clique when eliminating a vertex. As noted, we plan to accomplish this goal using randomized sampling. This section explains how to perform this task.

8.4.1 Setup

Let $\boldsymbol{S}$ be the Laplacian of a weighted multigraph over the set V of vertices. Let F be the *support* of $\boldsymbol{S}$; that is, the subset of vertices where $\boldsymbol{S}$ has an incident edge.

We will make two strong assumptions. First, we will assume that each multiedge in $\boldsymbol{S}$ has bounded leverage with respect to the original Laplacian:

$$\text{For } e = uv \in \boldsymbol{S}, \quad w_{\boldsymbol{S}}(e)\, \varrho(u, v) \le \frac{1}{R}. \tag{8.4}$$

We will often subscript the Laplacian $\boldsymbol{S}$ to specify the multigraph. Second, we will assume that

$$\|\boldsymbol{\Phi}(\boldsymbol{S})\| \le 2. \tag{8.5}$$

In other words, the entire multigraph specified by $\boldsymbol{S}$ has bounded leverage with respect to the target Laplacian $\boldsymbol{L}$.

The Laplacian $\boldsymbol{S}$ evolves as the SparseCholesky algorithm progresses. We will ensure that these two properties hold, by force if necessary.

8.4.2 Eliminating a vertex

Fix a vertex u to eliminate from the Laplacian $\boldsymbol{S}$. To do so, we first construct the star induced by the vertex:

$$\textsc{star}(u, \boldsymbol{S}) = \sum\nolimits_{e=uv\in\boldsymbol{S}} w_{\boldsymbol{S}}(e)\,\boldsymbol{\Delta}_e.$$

That is, the star contains all of the multiedges in $\boldsymbol{S}$ that are incident on u. Recall that $\deg(u, \boldsymbol{S})$ is the number of multiedges incident on u in $\boldsymbol{S}$, i.e., the cardinality of the star. The total weight of the vertex is

$$w_{\boldsymbol{S}}(u) = \sum\nolimits_{e\in\textsc{star}(u,\boldsymbol{S})} w_{\boldsymbol{S}}(e).$$

The clique induced by u has the Laplacian matrix

$$\textsc{clique}(u, \boldsymbol{S}) = \sum_{\substack{e_1=uv_1\in\textsc{star}(u,\boldsymbol{S})\\ e_2=uv_2\in\textsc{star}(u,\boldsymbol{S})}} \frac{w_{\boldsymbol{S}}(e_1)\,w_{\boldsymbol{S}}(e_2)}{2w_{\boldsymbol{S}}(u)}\,\boldsymbol{\Delta}_{v_1v_2}.$$

Recall that each multiedge e in the star appears once in each sum, so the total number of multiedges in the clique is $\deg_{\boldsymbol{S}}(u)^2$.

Our project is to construct a Laplacian matrix $\boldsymbol{K}$ that serves as an unbiased estimator for the clique:

$$\mathbb{E}[\boldsymbol{K} \mid u] = \textsc{clique}(u, \boldsymbol{S}).$$

We will insist that each multiedge in the approximation $\boldsymbol{K}$ has the form v_1v_2 where the multiedges $e_1 = uv_1$ and $e_2 = uv_2$ both appear in $\textsc{star}(u, \boldsymbol{S})$. Moreover, the total number of multiedges in $\boldsymbol{K}$ will not exceed $\deg_{\boldsymbol{S}}(u)$, the number of multiedges in the star that we remove. This is a quadratic reduction in complexity!

8.4.3 The sampling procedure

We are now prepared to detail the method for constructing a sparse, unbiased estimator of the clique.

Summary of CliqueSample:

1. Construct a probability mass on the multiedges in the star:

$$p(e) = \frac{w_{\boldsymbol{S}}(e)}{w_{\boldsymbol{S}}(u)} \quad \text{for each } e \in \textsc{star}(u, \boldsymbol{S}).$$

2. Draw a random multiedge $e_1 = uv_1$ from the multiedges in $\textsc{star}(u, \boldsymbol{S})$ according to the probability mass $\boldsymbol{p}$.
3. Draw a second random multiedge $e_2 = uv_2$ from the multiedges in $\textsc{star}(u, \boldsymbol{S})$ according to the uniform distribution.

4. Form the random Laplacian matrix of a new multiedge:

$$\boldsymbol{X} = \frac{w_{\boldsymbol{S}}(e_1)\, w_{\boldsymbol{S}}(e_2)}{w_{\boldsymbol{S}}(e_1) + w_{\boldsymbol{S}}(e_2)} \boldsymbol{\Delta}_{v_1 v_2}. \tag{8.6}$$

This construction has several important features that we will explore in the next paragraphs.

For now, we remark that this sampling procedure is analogous to the other matrix sampling approximations that we have discussed throughout the course. The closest parallel, naturally, is with the sparse graph approximation in Lecture 5. In that context, we sampled edges in proportion to their leverages. In the present context, we do not know the leverage scores. Instead, we exploit the fact that the effective resistances satisfy a triangle inequality to obtain adequate sampling probabilities.

8.4.4 Expectation of the random multiedge

First, let us compute the expectation of the Laplacian $\boldsymbol{X}$ of a random multiedge. We will see that the random multiedge is an unbiased estimator of the clique, up to a fixed scale factor.

Proposition 8.1 (Expectation of random Laplacian). The Laplacian $\boldsymbol{X}$ of the random multiedge (8.6) satisfies

$$\mathbb{E}\, \boldsymbol{X} = \frac{1}{\deg_{\boldsymbol{S}}(u)} \cdot \textsc{clique}(u, \boldsymbol{S}).$$

Proof. This result follows by direct calculation. Below, each of the sums iterates over the multiedges in $\textsc{star}(u, \boldsymbol{S})$, which we omit from the notation.

$$\begin{aligned}
\mathbb{E}\, \boldsymbol{X} &= \sum_{e_1 = uv_1} \frac{w_{\boldsymbol{S}}(e_1)}{w_{\boldsymbol{S}}(u)} \sum_{e_2 = uv_2} \frac{1}{\deg_{\boldsymbol{S}}(u, \boldsymbol{S})} \cdot \frac{w_{\boldsymbol{S}}(e_1)\, w_{\boldsymbol{S}}(e_2)}{w_{\boldsymbol{S}}(e_1) + w_{\boldsymbol{S}}(e_2)} \boldsymbol{\Delta}_{v_1 v_2} \\
&= \frac{1}{\deg_{\boldsymbol{S}}(u)} \sum_{\substack{e_1 = uv_1 \\ e_2 = uv_2}} \frac{w_{\boldsymbol{S}}(e_1)\, w_{\boldsymbol{S}}(e_2)}{w_{\boldsymbol{S}}(u)} \cdot \frac{w_{\boldsymbol{S}}(e_1)}{w_{\boldsymbol{S}}(e_1) + w_{\boldsymbol{S}}(e_2)} \boldsymbol{\Delta}_{v_1 v_2} \\
&= \frac{1}{\deg_{\boldsymbol{S}}(u)} \sum_{\substack{e_1 = uv_1 \\ e_2 = uv_2}} \frac{w_{\boldsymbol{S}}(e_1)\, w_{\boldsymbol{S}}(e_2)}{2 w_{\boldsymbol{S}}(u)} \boldsymbol{\Delta}_{v_1 v_2} = \frac{\textsc{clique}(u, \boldsymbol{S})}{\deg_{\boldsymbol{S}}(u)}.
\end{aligned}$$

The passage to the last line follows from the symmetry of the summands with respect to v_1 and v_2. ∎

8.4.5 Each multiedge has bounded leverage

Next, let us verify that the multiedge $\boldsymbol{X}$ constructed in (8.6) still has bounded leverage.

Proposition 8.2 (Bounded leverage). The random Laplacian matrix $\boldsymbol{X}$ defined in (8.6) satisfies the uniform bound

$$\|\boldsymbol{\Phi}(\boldsymbol{X})\| \le \frac{1}{R}.$$

Equivalently, given multiedges $e_1 = uv_1$ and $e_2 = uv_2$, the multiedge $e = v_1v_2$ with weight

$$w_e = \frac{w_{\boldsymbol{S}}(e_1)\, w_{\boldsymbol{S}}(e_2)}{w_{\boldsymbol{S}}(e_1) + w_{\boldsymbol{S}}(e_2)}$$

has leverage score

$$w_e \, \varrho(v_1, v_2) \le \frac{1}{R}.$$

Proof. This result is a consequence of the triangle inequality for effective resistances, Theorem 5.2. Indeed,

$$\begin{aligned} w_{\boldsymbol{S}}(e_1)\, w_{\boldsymbol{S}}(e_2) \cdot \varrho(v_1, v_2) & \\ &\le w_{\boldsymbol{S}}(e_2) \cdot w_{\boldsymbol{S}}(e_1)\, \varrho(u, v_1) + w_{\boldsymbol{S}}(e_1) \cdot w_{\boldsymbol{S}}(e_2)\, \varrho(u, v_2) \\ &\le \frac{1}{R}\big[w_{\boldsymbol{S}}(e_1) + w_{\boldsymbol{S}}(e_2)\big]. \end{aligned}$$

The last inequality holds because the weighted multiedges in $\boldsymbol{S}$ satisfy the uniform bound (8.4). Divide through by the bracket and identify the multiedge $e = v_1v_2$ with weight w_e to arrive at the stated result. ∎

8.4.6 Corrector for the random multiedge

We are now prepared to bound the corrector for the Laplacian $\boldsymbol{X}$ of the random multiedge (8.6). First, we center and normalize the random matrix. The result is then an immediate application of the Bernstein corrector construction, Proposition 7.8.

Proposition 8.3 (Corrector of random multiedge). Fix a vertex u. The random matrix $\boldsymbol{\Phi}(\boldsymbol{X} - \mathbb{E}\,\boldsymbol{X})$ admits the corrector

$$g(\theta) \cdot \frac{\boldsymbol{\Phi}(\textsc{clique}(u, \boldsymbol{S}))}{\deg_{\boldsymbol{S}}(u)} \quad \text{where} \quad g(\theta) = \frac{\theta^2/(2R)}{1 - |\theta|/(3R)}.$$

The random matrix $\boldsymbol{X}$ is defined in (8.6). Let us emphasize that the vertex u is not random at this stage.

Proof. Proposition 8.1 implies that the random matrix $\boldsymbol{\Phi}(\boldsymbol{X} - \mathbb{E}\,\boldsymbol{X})$ has mean zero. We have the uniform norm bound

$$\begin{aligned} \|\boldsymbol{\Phi}(\boldsymbol{X} - \mathbb{E}\,\boldsymbol{X})\| &= \|\boldsymbol{\Phi}(\boldsymbol{X}) - \boldsymbol{\Phi}(\mathbb{E}\,\boldsymbol{X})\| \\ &= \max\{\|\boldsymbol{\Phi}(\boldsymbol{X})\|,\ \|\boldsymbol{\Phi}(\mathbb{E}\,\boldsymbol{X})\|\} \le \|\boldsymbol{\Phi}(\boldsymbol{X})\| \le \frac{1}{R}. \end{aligned}$$

Since $\boldsymbol{X}$ is psd, so are $\boldsymbol{\Phi}(\boldsymbol{X})$ and $\boldsymbol{\Phi}(\mathbb{E}\,\boldsymbol{X})$. This justifies the norm identity. The first inequality is Jensen's. The second inequality is Proposition 8.2.

Let us compute the variance:

$$\begin{aligned} \mathbb{E}\,\boldsymbol{\Phi}(\boldsymbol{X} - \mathbb{E}\,\boldsymbol{X})^2 = \mathbb{E}\,\boldsymbol{\Phi}(\boldsymbol{X})^2 - \boldsymbol{\Phi}(\mathbb{E}\,\boldsymbol{X})^2 &\preccurlyeq \mathbb{E}\,\boldsymbol{\Phi}(\boldsymbol{X})^2 \\ &\preccurlyeq \mathbb{E}\left[\|\boldsymbol{\Phi}(\boldsymbol{X})\| \cdot \boldsymbol{\Phi}(\boldsymbol{X})\right] \preccurlyeq \frac{1}{R}\boldsymbol{\Phi}(\mathbb{E}\,\boldsymbol{X}) \\ &= \frac{\boldsymbol{\Phi}(\textsc{clique}(u, \boldsymbol{S}))}{R \deg_{\boldsymbol{S}}(u)}. \end{aligned}$$

We have repeatedly used the fact that $\mathbf{\Phi}$ is a positive linear map. The last identity is Proposition 8.1.

We now arrive at the result as a consequence of the Bernstein corrector bound, Proposition 7.8, and a scaling argument. ∎

8.4.7 An unbiased estimator for the clique

Next, we construct an estimator for the clique induced by eliminating the fixed vertex u from the Laplacian $\boldsymbol{S}$. To do so, add up $\deg_{\boldsymbol{S}}(u)$ independent copies of the random Laplacian $\boldsymbol{X}$ defined in (8.6):

$$\boldsymbol{K} = \sum\nolimits_{j=1}^{\deg_{\boldsymbol{S}}(u)} \boldsymbol{X}_j \quad \text{where } \boldsymbol{X}_j \sim \boldsymbol{X} \text{ iid.} \tag{8.7}$$

Since Laplacians form a convex cone, $\boldsymbol{K}$ is also the Laplacian of a multigraph. Let us verify that $\boldsymbol{K}$ is an unbiased estimator of the clique and compute its corrector.

Proposition 8.4 (Corrector of clique estimator: Fixed vertex). Fix a vertex u. The random matrix $\boldsymbol{K}$ defined in (8.7) is an unbiased estimator of the clique induced by eliminating u from $\boldsymbol{S}$:

$$\mathbb{E}\,\boldsymbol{K} = \textsc{clique}(u, \boldsymbol{S}).$$

The centered random matrix $\mathbf{\Phi}(\boldsymbol{K} - \mathbb{E}\,\boldsymbol{K})$ has corrector

$$g(\theta) \cdot \mathbf{\Phi}(\textsc{clique}(u, \boldsymbol{S})) \quad \text{with} \quad g(\theta) = \frac{\theta^2/(2R)}{1 - |\theta|/(3R)}.$$

As before, we treat the vertex u as nonrandom.

Proof. The centered clique estimator decomposes as an independent sum:

$$\boldsymbol{K} - \mathbb{E}\,\boldsymbol{K} = \sum\nolimits_{i=1}^{\deg_{\boldsymbol{S}}(u)} (\boldsymbol{X}_i - \mathbb{E}\,\boldsymbol{X}_i).$$

Proposition 7.11 states that the corrector tensorizes (over an independent sum). The result follows from Proposition 8.3. ∎

8.4.8 The clique induced by a random vertex

To complete our analysis of clique sampling, we consider what happens when we draw the vertex u at random.

First, let us develop several properties of the clique induced by an arbitrary vertex u. Recall from (8.3) that $\textsc{clique}(u, \boldsymbol{S})$ is a Laplacian matrix obtained by subtracting a psd matrix from $\textsc{star}(u, \boldsymbol{S})$. Moreover, $\textsc{star}(u, \boldsymbol{S})$ is the Laplacian of a subset of multiedges in $\boldsymbol{S}$. Therefore,

$$\mathbf{0} \preccurlyeq \textsc{clique}(u, \boldsymbol{S}) \preccurlyeq \textsc{star}(u, \boldsymbol{S}) \preccurlyeq \boldsymbol{S}.$$

Using the assumption (8.5), we obtain the bound

$$\|\mathbf{\Phi}(\textsc{clique}(u, \boldsymbol{S}))\| \le \|\mathbf{\Phi}(\boldsymbol{S})\| \le 2.$$

In other words, the whole clique has bounded leverage.

Second, we compute the average of the clique with respect to a vertex u drawn uniformly from the support F of the Laplacian $\boldsymbol{S}$. Note that

$$\begin{aligned}\mathbb{E}_u \operatorname{CLIQUE}(u, \boldsymbol{S}) \preccurlyeq \mathbb{E}_u \operatorname{STAR}(u, \boldsymbol{S}) &= \frac{1}{|\mathsf{F}|} \cdot \sum\nolimits_{u \in \mathsf{F}} \sum\nolimits_{e \in \operatorname{STAR}(u, \boldsymbol{S})} w_{\boldsymbol{S}}(e)\, \boldsymbol{\Delta}_e \\ &= \frac{2}{|\mathsf{F}|} \sum\nolimits_{e \in \boldsymbol{S}} w_{\boldsymbol{S}}(e)\, \boldsymbol{\Delta}_e = \frac{2}{|\mathsf{F}|} \cdot \boldsymbol{S}.\end{aligned}$$

Indeed, every multiedge in $\boldsymbol{S}$ appears twice in the sum because we touch each of its two endpoints as we loop over the vertices in the support F of $\boldsymbol{S}$. Applying the normalizing map,

$$\mathbb{E}_u \boldsymbol{\Phi}(\operatorname{CLIQUE}(u, \boldsymbol{S})) \preccurlyeq \frac{2}{|\mathsf{F}|} \cdot \boldsymbol{\Phi}(\boldsymbol{S}) \preccurlyeq \frac{4}{|\mathsf{F}|} \cdot \mathbf{I}.$$

The last inequality requires the assumption (8.5).

8.4.9 Corrector for the clique estimator

With these results at hand, we can find a corrector for the clique estimator $\boldsymbol{K}$ for a randomly chosen vertex u.

Theorem 8.5 (Corrector for clique estimator). Let $\boldsymbol{S}$ be a multigraph supported on the vertex set F. Assume that the properties (8.4) and (8.5) hold. Draw u uniformly at random from F, and let $\boldsymbol{K}$ be the random estimator (8.7) for the clique induced by u. Then the random matrix $\boldsymbol{\Phi}(\boldsymbol{K} - \mathbb{E}[\boldsymbol{K} \mid u])$ has corrector

$$\frac{2\, f(\theta)}{|\mathsf{F}|} \cdot \mathbf{I} \quad \text{where} \quad f(\theta) = \exp\left(\frac{\theta^2/R}{1 - |\theta|/(3R)}\right) - 1.$$

The corrector is computed with respect to the randomness in the summands $\boldsymbol{X}_i$ and in the vertex u.

Proof. Proposition 8.4 gives a corrector of $\Phi(\boldsymbol{K} - \mathbb{E}[\boldsymbol{K} \mid u])$ with respect to the randomness in the summands $\boldsymbol{X}_i$. This corrector is

$$g(\theta)\, \boldsymbol{\Phi}(\operatorname{CLIQUE}(v, \boldsymbol{S})) \quad \text{where} \quad g(\theta) = \frac{\theta^2/(2R)}{1 - |\theta|/(3R)}.$$

We have shown that

$$\mathbb{E}_u \boldsymbol{\Phi}(\operatorname{CLIQUE}(u, \boldsymbol{S})) \preccurlyeq \frac{4}{|\mathsf{F}|} \cdot \mathbf{I} \quad \text{and} \quad \|\boldsymbol{\Phi}(\operatorname{CLIQUE}(u, \boldsymbol{S}))\| \le 2.$$

Therefore, with respect to the random choice of the vertex u, the random matrix $\boldsymbol{\Phi}(\operatorname{CLIQUE}(u, \boldsymbol{S}))$ admits the corrector

$$\frac{4h(\theta)}{|\mathsf{F}|} \cdot \mathbf{I} \quad \text{where} \quad h(\theta) = \frac{\mathrm{e}^{2\theta} - 1}{2}.$$

This is just the Chernoff corrector bound, Proposition 7.9. The result follows from the composition rule, Proposition 7.12, since $f = 2\,(h \circ g)$. ∎

8.5 Analysis of SparseCholesky

We are finally prepared to establish that the SparseCholesky algorithm succeeds when we use the clique sampling procedure developed in Section 8.4. Fix a parameter $\varepsilon \in (0, 1)$. Our first goal is to prove that, with high probability,

$$\|\boldsymbol{\Phi}(\boldsymbol{L}_n - \boldsymbol{L}_0)\| \leq \varepsilon.$$

Afterward, we must argue that the runtime of the algorithm is controlled.

8.5.1 A stopping time

It is sufficient to show that, with high probability,

$$\max_{i=0,\dots,n} \|\boldsymbol{\Phi}(\boldsymbol{L}_i - \boldsymbol{L}_0)\| \leq \varepsilon.$$

Let us define the stopping time

$$T = \min\{0 \leq i \leq n : \|\boldsymbol{\Phi}(\boldsymbol{L}_i - \boldsymbol{L}_0)\| > \varepsilon\}.$$

If this event never occurs, then $T = +\infty$. For each $i < T$, observe that $\|\boldsymbol{\Phi}(\boldsymbol{L}_i)\| \leq 1 + \varepsilon$ because $\|\boldsymbol{\Phi}(\boldsymbol{L}_0)\| = 1$.

We will consider the stopped martingale

$$\boldsymbol{Y}_i = \boldsymbol{\Phi}(\boldsymbol{L}_{i\wedge T} - \boldsymbol{L}_0).$$

Clearly, it suffices to obtain a probability bound for the event that the stopped martingale exhibits a large deviation:

$$\max_{0\leq i\leq n} \|\boldsymbol{Y}_i\| > \varepsilon.$$

We will treat the maximum and minimum eigenvalue parts of this spectral norm bound separately, but the arguments are symmetrical.

8.5.2 The approximate Schur complements

The purpose of introducing the stopped martingale is to guarantee that the approximate Schur complements are uniformly bounded. Indeed, since $\mathbf{0} \preccurlyeq \boldsymbol{S}_i \preccurlyeq \boldsymbol{L}_i$ for each i, we have the consequence that

$$\max_{i\leq T} \|\boldsymbol{\Phi}(\boldsymbol{S}_{i-1})\| \leq \max_{i<T} \|\boldsymbol{\Phi}(\boldsymbol{L}_i)\| \leq 2.$$

This condition delivers the uniform bound (8.5), irrespective of the choice of ε.

Moreover, the initial Laplacian $\boldsymbol{S}_0$ consists of multiedges with leverage score bounded by $1/R$. At each step of the iteration, we remove some multiedges from $\boldsymbol{S}_{i-1}$ and then add back a random clique estimator. Proposition 8.2 ensures that each multiedge in the clique estimator also has leverage score bounded by $1/R$. By induction, the assumption (8.4) holds in every iteration.

8.5.3 The corrector process

For $i \geq 1$, the difference sequence of the martingale $\boldsymbol{Y}_i$ is

$$\boldsymbol{Y}_i - \boldsymbol{Y}_{i-1} = \begin{cases} \boldsymbol{\Phi}(\boldsymbol{K}_i - \mathbb{E}_{i-1}[\boldsymbol{K}_i \,|\, u_i]), & i \leq T \\ \mathbf{0}, & i > T. \end{cases}$$

For $i \leq T$, the matrix $\boldsymbol{S}_{i-1}$ satisfies the conditions required to invoke Theorem 8.5. The support of $\boldsymbol{S}_{i-1}$ has cardinality $|\mathsf{F}_{i-1}| = n - i + 1$. Therefore, the increment $\boldsymbol{Y}_i - \boldsymbol{Y}_{i-1}$ has the corrector

$$\frac{2g(\theta)}{n-i+1} \cdot \mathbf{I} \quad \text{where} \quad g(\theta) = \exp\left(\frac{\theta^2/R}{1 - |\theta|/(3R)}\right) - 1.$$

(This is computed conditional on all of the random choices up to step $i-1$.) For $i > T$, we can take the corrector to be the zero matrix.

Therefore, owing to Proposition 7.10, the martingale admits the nonrandom corrector process

$$g\,\boldsymbol{W}_i = 2g\left[\sum\nolimits_{j=1}^{i\wedge T} \frac{1}{n-j+1}\right] \cdot \mathbf{I} \preccurlyeq 2g\,\log(\mathrm{e}n) \cdot \mathbf{I}.$$

To obtain the bound, we have summed the harmonic series up to $j = n$.

The same corrector is valid for the negation $\{-\boldsymbol{Y}_i\}$ of the martingale, so we can obtain matching bounds for the maximum and minimum eigenvalues.

8.5.4 The martingale tail bound

Finally, we can bound the probability that the Laplacian martingale exhibits a large deviation. Set $\sigma^2 = 2\log(\mathrm{e}n)$.

$$\begin{aligned} \mathbb{P}\left\{\|\boldsymbol{\Phi}(\boldsymbol{L}_n - \boldsymbol{L}_0)\| > \varepsilon\right\} &\leq \mathbb{P}\left\{\exists i : \|\boldsymbol{\Phi}(\boldsymbol{L}_i - \boldsymbol{L}_0)\| > \varepsilon\right\} \\ &= \mathbb{P}\left\{\exists i : \|\boldsymbol{Y}_i\| > \varepsilon\right\} \\ &\leq \mathbb{P}\left\{\exists i : \lambda_{\max}(\boldsymbol{Y}_i) \geq \varepsilon\right\} + \mathbb{P}\left\{\exists i : \lambda_{\max}(-\boldsymbol{Y}_i) \geq \varepsilon\right\}. \end{aligned}$$

Indeed, the stopping time is triggered by the failure event, so we can pass to the stopped martingale. Then we split the spectral norm into eigenvalues so we can apply the master tail bound for matrix martingales.

We are in a good position to bound the last two probabilities.

$$\begin{aligned} \mathbb{P}\left\{\exists i : \lambda_{\max}(\boldsymbol{Y}_i) \geq \varepsilon\right\} &= \mathbb{P}\left\{\exists i : \lambda_{\max}(\boldsymbol{Y}_i) \geq \varepsilon \text{ and } \lambda_{\max}(\boldsymbol{W}_i) \leq \sigma^2\right\} \\ &\leq n \cdot \inf_{\theta>0} \exp\left(-\varepsilon\theta + g(\theta)\,\sigma^2\right). \end{aligned}$$

We have used the fact that $\lambda_{\max}(\boldsymbol{W}_i) \leq \sigma^2$ always. The last inequality is a direct application of Theorem 7.4, the master tail bound for matrix martingales. Likewise,

$$\mathbb{P}\left\{\exists i : \lambda_{\max}(-\boldsymbol{Y}_i) \geq \varepsilon\right\} \leq n \cdot \inf_{\theta>0} \exp\left(-\varepsilon\theta + g(\theta)\,\sigma^2\right).$$

This bound also follows from Theorem 7.4.

Altogether, we determine that

$$\mathbb{P}\left\{\|\boldsymbol{\Phi}(\boldsymbol{L}_n - \boldsymbol{L}_0)\| > \varepsilon\right\} \le 2n \cdot \inf_{\theta>0} \exp\big(-\varepsilon\theta + 2g(\theta)\log(\mathrm{e}n)\big).$$

We may select the parameters

$$\theta = 2\varepsilon^{-1}\log(\mathrm{e}n) \quad \text{and} \quad R = \lceil 4\theta^2 \rceil = \lceil 16\varepsilon^{-2}\log^2(\mathrm{e}n) \rceil.$$

In this case, $g(\theta) \le 0.3$ for any $n \ge 1$ and $\varepsilon < 1$. We have the overall bound

$$\mathbb{P}\left\{\|\boldsymbol{\Phi}(\boldsymbol{L}_n - \boldsymbol{L}_0)\| > \varepsilon\right\} \le (\mathrm{e}n)^{-0.4}.$$

The probability bound can, of course, be improved by increasing the value of θ. To do so, however, we must also increase the value of R, which means that we split the multiedges in the initial graph into more pieces.

8.5.5 The running time

Last, we must assess the running time of the SparseCholesky algorithm. The first step is to split the edges in the multigraph into R pieces to obtain a total of $M = Rm$ multiedges. This step costs $O(M)$ time and memory accesses.

At the outset, there are $M = Rm$ multiedges in the graph. At each iteration, we eliminate a vertex by removing all the multiedges incident on that vertex and adding a clique with (at most) the same number of multiedges. As a result, the number of multiedges in the graph never increases above M at any iteration.

Now, in iteration i, we select a vertex u_i at random from the $n - i + 1$ remaining vertices. In expectation, the number t_i of multiedges incident on u_i satisfies $t_i \le M/(n - i + 1)$.

To sample the clique induced by u_i, we need to draw t_i samples from a probability mass on t_i points. This task can be accomplished in $O(t_i)$ time overall [BP17]. The rest of the computation of the clique estimator and its introduction into the current Laplacian involve $O(t_i)$ arithmetic and memory accesses.

Afterward, we remove the star induced by u_i, which also contains t_i multiedges. This operation involves $O(t_i)$ arithmetic and memory accesses.

In summary, the expected running time of the algorithm is on the order of

$$M + \sum\nolimits_{i=1}^{n} t_i \asymp Rm \sum\nolimits_{i=1}^{n} \frac{1}{n - i + 1} \asymp Rm \log n.$$

To obtain an error of $\varepsilon = 0.5$, we can take $R = \Theta(\log^2 n)$. Therefore, the overall runtime is $O(m \log^3(n))$, in expectation.

8.5.6 The grand finale

To summarize, we have established the following result.

Theorem 8.6 (Sparse Cholesky). Let $\boldsymbol{L} \in \mathbb{H}_{\mathsf{V}}$ be the Laplacian matrix of a connected graph G on a set V of n vertices and with m weighted edges. The SparseCholesky algorithm produces a a morally lower-triangular matrix $\boldsymbol{C} \in \mathbb{R}^{\mathsf{V}\times\mathsf{V}}$ that satisfies

$$0.5\,\boldsymbol{L} \preccurlyeq \boldsymbol{C}\boldsymbol{C}^* \preccurlyeq 1.5\,\boldsymbol{L}.$$

> The matrix $\boldsymbol{C}$ has $O(m \log^2 n)$ nonzero entries. The expected running time is $O(m \log^3(n))$.

In view of our discussion of preconditioned conjugate gradient, we arrive at an algorithmic approach for solving Laplacian linear systems.

Corollary 8.7 (Laplacian systems). Given the preconditioner $\boldsymbol{L}$ computed by the Sparse-Cholesky algorithm, we can solve every consistent linear system in a graph Laplacian to relative error ε in the Dirichlet energy norm in time $O(m \log^2(n) \log(1/\varepsilon))$.

This is what we promised to prove. ■

Gutenberg Bible, British Library, public domain

Further Reading

Here is an incomplete collection of sources where you can learn more about contemporary random matrix theory and its applications.

Matrix concentration inequalities

The papers below develop results on how much a random matrix deviates from its mean in spectral norm. An important characteristic of these results is that they apply to a wide range of different types of random matrices, the constants are explicit (and reasonable), and the bounds are nonasymptotic.

- [Tro12] Joel A. Tropp, "User-friendly tail bounds for sums of random matrices."

  This is a foundational paper that develops the modern approach to matrix concentration via the subadditivity of matrix cumulants. It contains a complete catalog of exponential inequalities for an independent sum of random matrices.

- [Tro11a] Joel A. Tropp, "Freedman's inequality for matrix martingales."

  This foundational paper is a follow-up to [Tro12] that develops the approach to matrix martingales using corrector processes. It was inspired by Roberto Oliveira's paper [Oli10], which established a weaker version of the matrix Freedman inequality.

- [Tro15] Joel A. Tropp, "An introduction to matrix concentration inequalities."

  My monograph gives a thorough introduction to matrix concentration for independent sums, including many applications, a complete proof of Lieb's theorem, and an annotated bibliography of works on matrix concentration.

- [Tro16] Joel A. Tropp, "The expected norm of a sum of independent random matrices: An elementary approach."

The matrix Rosenthal inequalities are moment inequalities for a sum of independent random matrices that strengthen the matrix Bernstein inequality. This paper proves the matrix Rosenthal inequalities using elementary arguments.

- [Mac+14] Lester Mackey et al., "Matrix concentration inequalities via the method of exchangeable pairs."

  This paper develops another approach to matrix concentration using Stein's method. This approach is also more elementary than the approach using Lieb's theorem, and it applies to some types of random matrices that are more general than independent sums or martingales.

- [PMT16] Daniel Paulin et al., "Efron–Stein inequalities for random matrices."

  This paper shows how to use Stein's method to prove concentration inequalities for a matrix-valued function of independent random variables. This is potentially a very powerful result, but it has seen relatively few applications so far.

- [BH16] Afonso Bandeira and Ramon van Handel, "Sharp nonasymptotic bounds on the norm of random matrices with independent entries."

  This paper gives sharp bounds on the norm of a random matrix with independent entries, which is one of the most important random matrix models.

Lower tail inequalities

The minimum eigenvalue of a sum of psd random matrices exhibits a totally different kind of behavior from the maximum eigenvalue. The following papers tackle this important problem.

- [KM15] Vladimir Koltchinskii and Shahar Mendelson, "Bounding the smallest singular value of a random matrix without concentration."

  This paper explains how to use the small-ball method to bound the smallest singular value of a random matrix with independent rows.

- [Tro16] Joel A. Tropp, "Convex recovery of a structured signal from independent random measurements."

  This expository work gives a simplified account of the small-ball method for controlling the minimum (conic) singular value of a random matrix with independent rows.

- [Oli16] Roberto Oliveira, "The lower tail of random quadratic forms with applications to ordinary least squares."

  This paper proves a lower tail inequality for a sum of independent, random psd matrices. The method is fascinating and strikingly different from other approaches.

High-dimensional probability

Here are some surveys that I have found useful. They cover various aspects of high-dimensional probability with applications to modern random matrix theory.

- [FR13] Simon Foucart and Holger Rauhut, *A Mathematical Introduction to Compressive Sensing*.

This book develops a collection of methods for studying structured random matrices, and it proceeds from first principles.

- [Han16] Ramon van Handel, *Probability in High Dimensions*, 2016.

  These lecture notes contain a sophisticated mathematical treatment of high-dimensional probability, including some applications to random matrix theory.

- [Han17] Ramon van Handel, "Structured random matrices."

  This accessible survey covers some very recent results about structured random matrices.

- [Ver18] Roman Vershynin, *High-Dimensional Probability*.

  This textbook gives an elementary introduction to the methods of high-dimensional probability, including some random matrix theory and many applications in data science.

Classical random matrix theory

Last, we mention a few resources for learning about the more established parts of random matrix theory.

- [Tao12] Terence Tao, *Topics in Random Matrix Theory*.

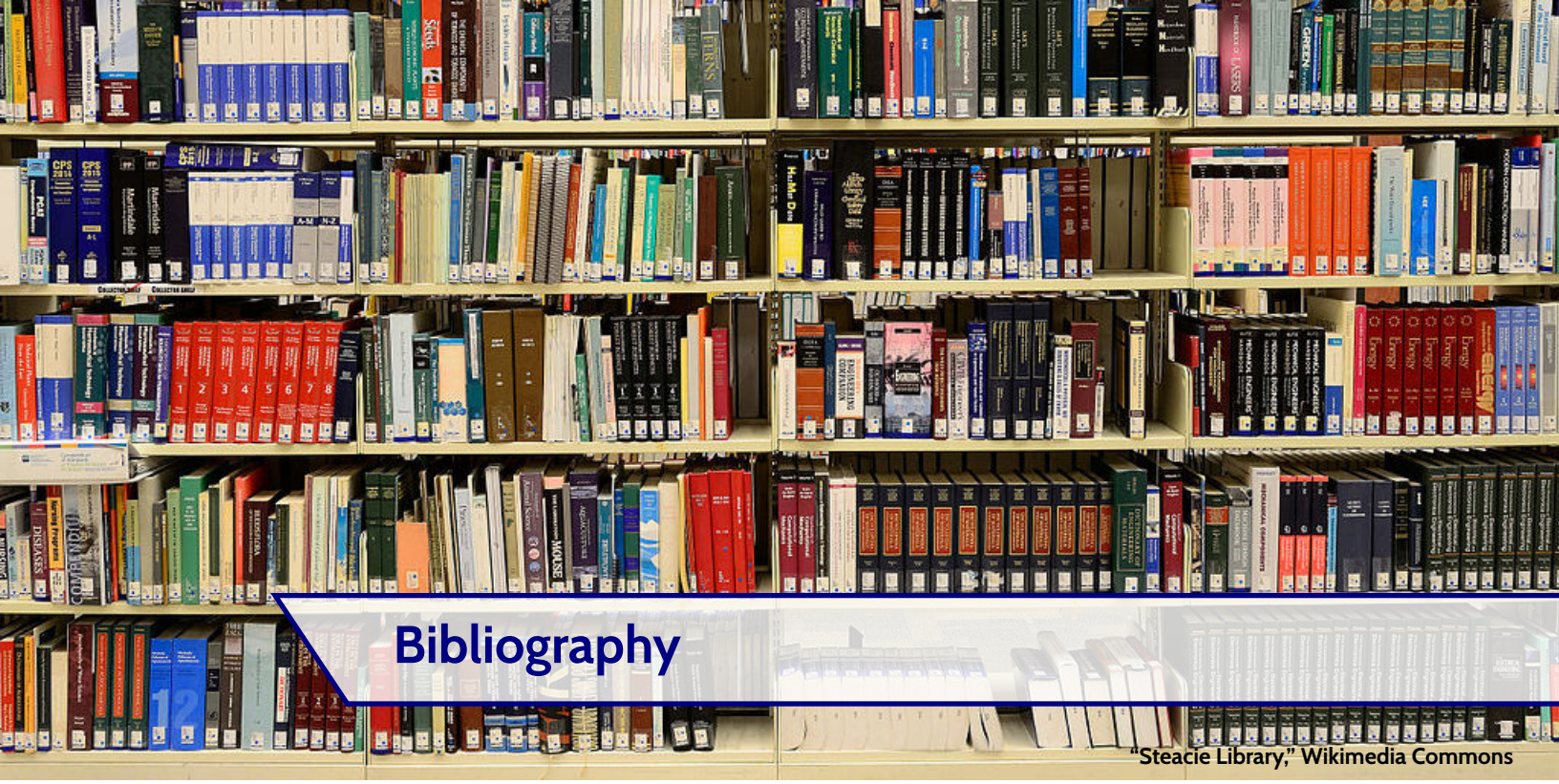
  This textbook gives an accessible introduction to the classical theory of random matrices.

- [Kem13] Todd Kemp, *Introduction to Random Matrix Theory*.

  These lecture notes provide another readable treatment of classical random matrix theory.

- [NS06] Alexandru Nica and Roland Speicher, *Lectures on the Combinatorics of Free Probability*.

  This monograph introduces the theory of free probability from a combinatorial point of view.



Bibliography

"Steacie Library," Wikimedia Commons

[AW02] Rudolf Ahlswede and Andreas Winter. "Strong converse for identification via quantum channels". In: *IEEE Trans. Inform. Theory* 48.3 (2002), pages 569–579. DOI: 10.1109/18.985947.

[Axl15] Sheldon Axler. *Linear algebra done right*. Third. Undergraduate Texts in Mathematics. Springer, Cham, 2015, pages xviii+340. DOI: 10.1007/978-3-319-11080-6.

[BH16] Afonso S. Bandeira and Ramon van Handel. "Sharp nonasymptotic bounds on the norm of random matrices with independent entries". In: *Ann. Probab.* 44.4 (2016), pages 2479–2506. DOI: 10.1214/15-AOP1025.

[BG13] Richard F. Bass and Karlheinz Gröchenig. "Relevant sampling of band-limited functions". In: *Illinois J. Math.* 57.1 (2013), pages 43–58. URL: http://projecteuclid.org/euclid.ijm/1403534485.

[BSS14] Joshua Batson, Daniel A. Spielman, and Nikhil Srivastava. "Twice-Ramanujan sparsifiers". In: *SIAM Rev.* 56.2 (2014), pages 315–334. DOI: 10.1137/130949117.

[Bha97] Rajendra Bhatia. *Matrix analysis*. Volume 169. Graduate Texts in Mathematics. Springer-Verlag, New York, 1997, pages xii+347. DOI: 10.1007/978-1-4612-0653-8. URL: http://dx.doi.org/10.1007/978-1-4612-0653-8.

[Bha07] Rajendra Bhatia. *Positive definite matrices*. Princeton Series in Applied Mathematics. Princeton University Press, Princeton, NJ, 2007, pages x+254.

[BLM13] Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. *Concentration inequalities*. A nonasymptotic theory of independence, With a foreword by Michel Ledoux. Oxford University Press, Oxford, 2013, pages x+481. DOI: 10.1093/acprof:oso/9780199535255.001.0001. URL: http://dx.doi.org/10.1093/acprof:oso/9780199535255.001.0001.

[BP17] Karl Bringmann and Konstantinos Panagiotou. "Efficient sampling methods for discrete distributions". In: *Algorithmica* 79.2 (2017), pages 484–508. DOI: 10.1007/s00453-016-0205-0.

[Buc01] Artur Buchholz. "Operator Khintchine inequality in non-commutative probability". In: *Math. Ann.* 319.1 (2001), pages 1–16. DOI: 10.1007/PL00004425. URL: http://dx.doi.org/10.1007/PL00004425.

[CC13] Xiaohong Chen and Timothy M. Christensen. "Optimal uniform convergence rates for sieve nonparametric instrumental variables regression". Available at http://arXiv.org/abs/1311.0412. Nov. 2013.

[Che+14] Yudong Chen et al. "Coherent matrix completion". In: *Proc. 31st Intl. Conf. Machine Learning*. Beijing, 2014.

[CSW12] Sin-Shuen Cheung, Anthony Man-Cho So, and Kuncheng Wang. "Linear matrix inequalities with stochastically dependent perturbations and applications to chance-constrained semidefinite optimization". In: *SIAM J. Optim.* 22.4 (2012), pages 1394–1430. DOI: 10.1137/110822906. URL: http://dx.doi.org/10.1137/110822906.

[CDL13] Albert Cohen, Mark A. Davenport, and Dany Leviatan. "On the stability and accuracy of least squares approximations". In: *Found. Comput. Math.* 13.5 (2013), pages 819–834. DOI: 10.1007/s10208-013-9142-3. URL: http://dx.doi.org/10.1007/s10208-013-9142-3.

[CG14] Paul Constantine and David Gleich. "Computing active subspaces". Available at http://arXiv.org/abs/1408.0545. Aug. 2014.

[DKC13] Josip Djolonga, Andreas Krause, and Volkan Cevher. "High-Dimensional Gaussian Process Bandits". In: *Advances in Neural Information Processing Systems 26*. Edited by C.J.C. Burges et al. Curran Associates, Inc., 2013, pages 1025–1033. URL: http://papers.nips.cc/paper/5152-high-dimensional-gaussian-process-bandits.pdf.

[FSV12] Massimo Fornasier, Karin Schnass, and Jan Vybiral. "Learning functions of few arbitrary linear parameters in high dimensions". In: *Found. Comput. Math.* 12.2 (2012), pages 229–262. DOI: 10.1007/s10208-012-9115-y. URL: http://dx.doi.org/10.1007/s10208-012-9115-y.

[FR13] Simon Foucart and Holger Rauhut. *A mathematical introduction to compressive sensing*. Applied and Numerical Harmonic Analysis. Birkhäuser/Springer, New York, 2013, pages xviii+625. DOI: 10.1007/978-0-8176-4948-7.

[Fre75] David A. Freedman. "On tail probabilities for martingales". In: *Ann. Probability* 3 (1975), pages 100–118. DOI: 10.1214/aop/1176996452.

[GN51] Herman H. Goldstine and John von Neumann. "Numerical inverting of matrices of high order. II". In: *Proc. Amer. Math. Soc.* 2 (1951), pages 188–202. DOI: 10.2307/2032484.

[Grc11] Joseph F. Grcar. "John von Neumann's analysis of Gaussian elimination and the origins of modern numerical analysis". In: *SIAM Rev.* 53.4 (2011), pages 607–682. DOI: 10.1137/080734716.

[Guh+18] Madeline Guha et al. "Fast state tomography with optimal error bounds". Available at http://arXiv.org/abs/1809.11162. Sept. 2018.

[Haa+17] Jidong Haah et al. "Sample-Optimal Tomography of Quantum States". In: *IEEE Transactions on Information Theory* 63.9 (Sept. 2017), pages 5628–5641. DOI: 10.1109/TIT.2017.2719044.

[Han16] Ramon van Handel. "Probability in High Dimensions". APC 550 Lecture Notes, Princeton University. Available at https://web.math.princeton.edu/~rvan/APC550.pdf. Dec. 2016.

[Han17] Ramon van Handel. "Structured random matrices". In: *Convexity and concentration*. Volume 161. IMA Vol. Math. Appl. Springer, New York, 2017, pages 107–156.

[HO14] Nicholas J. A. Harvey and Neil Olver. "Pipage Rounding, Pessimistic Estimators and Matrix Concentration". In: *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*. SODA '14. Portland, Oregon: SIAM, 2014, pages 926–945. URL: http://dl.acm.org/citation.cfm?id=2634074.2634143.

[Hig02] Nicholas J. Higham. *Accuracy and stability of numerical algorithms*. Second. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2002, pages xxx+680. DOI: 10.1137/1.9780898718027.

[Hol12] Alexander S. Holevo. *Quantum systems, channels, information*. Volume 16. De Gruyter Studies in Mathematical Physics. A mathematical introduction. De Gruyter, Berlin, 2012, pages xiv+349. DOI: 10.1515/9783110273403. URL: http://dx.doi.org/10.1515/9783110273403.

[HCG14] Qixing Huang, Yuxin Chen, and Leonidas Guibas. "Near-optimal joint object matching via convex relaxation". In: *Proc. 31st Intl. Conf. Machine Learning*. Beijing, 2014.

[KK12] Purushottam Kar and Harish Karnick. "Random Feature Maps for Dot Product Kernels". In: *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics*. Edited by Neil D. Lawrence and Mark Girolami. Volume 22. Proceedings of Machine Learning Research. La Palma, Canary Islands: PMLR, 21–23 Apr 2012, pages 583–591. URL: http://proceedings.mlr.press/v22/kar12.html.

[Kem13] Todd Kemp. "Math 247A: Introduction to random matrix theory". Available at http://www.math.ucsd.edu/~tkemp/247A/247A.Notes.pdf. 2013.

[Kol11] Vladimir Koltchinskii. *Oracle inequalities in empirical risk minimization and sparse recovery problems*. Volume 2033. Lecture Notes in Mathematics. Lectures from the 38th Probability Summer School held in Saint-Flour, 2008, École d'Été de Probabilités de Saint-Flour. [Saint-Flour Probability Summer School]. Springer, Heidelberg, 2011, pages x+254. DOI: 10.1007/978-3-642-22147-7. URL: http://dx.doi.org/10.1007/978-3-642-22147-7.

[KM15] Vladimir Koltchinskii and Shahar Mendelson. "Bounding the smallest singular value of a random matrix without concentration". In: *Int. Math. Res. Not. IMRN* 23 (2015), pages 12991–13008. DOI: 10.1093/imrn/rnv096.

[Kyn17] Rasmus Kyng. "Approximate Gaussian elimination". PhD thesis. Yale University, 2017, page 120.

[KS16] Rasmus Kyng and Sushant Sachdeva. "Approximate Gaussian elimination for Laplacians—fast, sparse, and simple". In: *57th Annual IEEE Symposium on Foundations of Computer Science—FOCS 2016*. IEEE Computer Soc., Los Alamitos, CA, 2016, pages 573–582.

[Kyn+16] Rasmus Kyng et al. "Sparsified Cholesky and multigrid solvers for connection Laplacians". In: *STOC'16—Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing*. ACM, New York, 2016, pages 842–850.

[Lie73] Elliott H. Lieb. "Convex trace functions and the Wigner-Yanase-Dyson conjecture". In: *Advances in Math.* 11 (1973), pages 267–288. DOI: 10.1016/0001-8708(73)90011-X.

[Lop+14] David Lopez-Paz et al. "Randomized nonlinear component analysis". In: *Proc. 31st Intl. Conf. Machine Learning*. Beijing, July 2014.

[Lus86] Françoise Lust-Piquard. "Inégalités de Khintchine dans C_p $(1 < p < \infty)$". In: *C. R. Acad. Sci. Paris Sér. I Math.* 303.7 (1986), pages 289–292.

[LP91] Françoise Lust-Piquard and Gilles Pisier. "Noncommutative Khintchine and Paley inequalities". In: *Ark. Mat.* 29.2 (1991), pages 241–260. DOI: 10.1007/BF02384340.

[Mac+14] Lester Mackey et al. "Matrix concentration inequalities via the method of exchangeable pairs". In: *Ann. Probab.* 42.3 (2014), pages 906–945. DOI: 10.1214/13-AOP892. URL: http://dx.doi.org/10.1214/13-AOP892.

[MB17] William B. March and George Biros. "Far-Field Compression for Fast Kernel Summation Methods in High Dimensions". In: *Appl. Comput. Harmon. Anal.* 43.1 (July 2017), pages 39–75. DOI: 10.1016/j.acha.2015.09.007.

[MKR12] Emilie Morvant, Sokol Koço, and Liva Ralaivola. "PAC-Bayesian Generalization Bound on Confusion Matrix for Multi-Class Classification". In: *Proc. 29th Intl. Conf. Machine Learning*. Edinburgh, 2012.

[NG47] John von Neumann and Herman H. Goldstine. "Numerical inverting of matrices of high order". In: *Bull. Amer. Math. Soc.* 53 (1947), pages 1021–1099. DOI: 10.1090/S0002-9904-1947-08909-6.

[NS06] Alexandru Nica and Roland Speicher. *Lectures on the combinatorics of free probability*. Volume 335. London Mathematical Society Lecture Note Series. Cambridge University Press, Cambridge, 2006, pages xvi+417. DOI: 10.1017/CBO9780511735127.

[Oli10] Roberto Imbuzeiro Oliveira. "The spectrum of random k-lifts of large graphs (with possibly large k)". In: *J. Comb.* 1.3-4 (2010), pages 285–306. DOI: 10.4310/JOC.2010.v1.n3.a2. URL: http://dx.doi.org/10.4310/JOC.2010.v1.n3.a2.

[Oli16] Roberto Imbuzeiro Oliveira. "The lower tail of random quadratic forms with applications to ordinary least squares". In: *Probab. Theory Related Fields* 166.3-4 (2016), pages 1175–1194. DOI: 10.1007/s00440-016-0738-9.

[PMT16] Daniel Paulin, Lester Mackey, and Joel A. Tropp. "Efron-Stein inequalities for random matrices". In: *Ann. Probab.* 44.5 (2016), pages 3431–3473. DOI: 10.1214/15-AOP1054. URL: https://doi-org.clsproxy.library.caltech.edu/10.1214/15-AOP1054.

[PX97] Gilles Pisier and Quanhua Xu. "Non-commutative martingale inequalities". In: *Comm. Math. Phys.* 189.3 (1997), pages 667–698. DOI: 10.1007/s002200050224.

[RR07] Ali Rahimi and Benjamin Recht. "Random features for large-scale kernel machines". In: *Adv. Neural Information Processing Systems*. Vancouver, 2007.

[Rud99] Mark Rudelson. "Random vectors in the isotropic position". In: *J. Funct. Anal.* 164.1 (1999), pages 60–72. DOI: 10.1006/jfan.1998.3384. URL: http://dx.doi.org/10.1006/jfan.1998.3384.

[SS01] Bernhard Schölkopf and Alex Smola. *Learning with kernels*. Adaptive Computation and Machine Learning series. MIT Press, 2001.

[Spi12] Daniel A. Spielman. "Spectral graph theory". In: *Combinatorial scientific computing*. Chapman & Hall/CRC Comput. Sci. Ser. CRC Press, Boca Raton, FL, 2012, pages 495–524. DOI: 10.1201/b11644-19.

[Spi] Daniel A. Spielman. *CPSC 662 / AMTH 561: Spectral Graph Theory*. URL: http://www.cs.yale.edu/homes/spielman/561/syllabus.html (visited on 06/29/2019).

[SS11] Daniel A. Spielman and Nikhil Srivastava. "Graph sparsification by effective resistances". In: *SIAM J. Comput.* 40.6 (2011), pages 1913–1926. DOI: 10.1137/080734029.

[Tao12] Terence Tao. *Topics in random matrix theory*. Volume 132. Graduate Studies in Mathematics. American Mathematical Society, Providence, RI, 2012, pages x+282.

[Tom74] Nicole Tomczak-Jaegermann. "The moduli of smoothness and convexity and the Rademacher averages of trace classes $S_p(1 \leq p < \infty)$". In: *Studia Math.* 50 (1974), pages 163–182.

[TB97] Lloyd N. Trefethen and David Bau III. *Numerical linear algebra*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1997, pages xii+361. DOI: 10.1137/1.9780898719574.

[Tro11a] Joel A. Tropp. "Freedman's inequality for matrix martingales". In: *Electron. Commun. Probab.* 16 (2011), pages 262–270. DOI: 10.1214/ECP.v16-1624.

[Tro11b] Joel A. Tropp. "Improved analysis of the subsampled randomized Hadamard transform". In: *Adv. Adapt. Data Anal.* 3.1-2 (2011), pages 115–126. DOI: 10.1142/S1793536911000787. URL: http://dx.doi.org/10.1142/S1793536911000787.

[Tro12] Joel A. Tropp. "User-friendly tail bounds for sums of random matrices". In: *Found. Comput. Math.* 12.4 (2012), pages 389–434. DOI: 10.1007/s10208-011-9099-z.

[Tro15] Joel A. Tropp. "An introduction to matrix concentration inequalities". In: *Foundations and Trends in Machine Learning* 8.1–2 (May 2015), pages 1–230.

[Tro16] Joel A. Tropp. "The expected norm of a sum of independent random matrices: an elementary approach". In: *High dimensional probability VII*. Volume 71. Progr. Probab. Springer, [Cham], 2016, pages 173–202. DOI: 10.1007/978-3-319-40519-3_8.

[Ver18] Roman Vershynin. *High-dimensional probability*. Volume 47. Cambridge Series in Statistical and Probabilistic Mathematics. An introduction with applications in data science, With a foreword by Sara van de Geer. Cambridge University Press, Cambridge, 2018, pages xiv+284. DOI: 10.1017/9781108231596.

[Wig] Yuval Wigderson. *Harmonic functions on graphs*. URL: http://web.stanford.edu/~yuvalwig/math/teaching/HarmonicNotes.pdf (visited on 06/29/2019).

[Wil91] David Williams. *Probability with martingales*. Cambridge Mathematical Textbooks. Cambridge University Press, Cambridge, 1991, pages xvi+251. DOI: 10.1017/CBO9780511813658.